

IBM Cúram Social Program Management



Cómo trabajar con Intelligent Evidence Gathering (IEG)

Versión 6.05

IBM Cúram Social Program Management



Cómo trabajar con Intelligent Evidence Gathering (IEG)

Versión 6.05

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 67

Revisado: mayo de 2013

Esta edición se aplica a IBM Cúram Social Program Management, versión 6.0 5 y a todos los releases posteriores a menos que se indique lo contrario en ediciones nuevas.

Materiales bajo licencia - Propiedad de IBM.

© Copyright IBM Corporation 2012, 2013.

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

Contenido

Figuras v

Tablas vii

Capítulo 1. Introducción 1

- 1.1 Público al que va dirigida 1
- 1.2 Finalidad 1
- 1.3 Lectura adicional. 1

Capítulo 2. Cómo empezar 3

- 2.1 Introducción 3
- 2.2 Acerca de IEG. 3
 - 2.2.1 Almacén de datos (DS) 3
 - 2.2.2 Almacén de recursos (RS) 4
 - 2.2.3 Estructura de script 4
- 2.3 Evaluación del uso de IEG 4
- 2.4 Aspectos básicos 5
 - 2.4.1 Crear un esquema. 5
 - 2.4.2 Crear un script 7
 - 2.4.3 Adición de una página de resumen a un script IEG 9
 - 2.4.4 Ejecutar un script 10

Capítulo 3. Captura de información de cliente 13

- 3.1 Introducción 13
- 3.2 Familias y unidades familiares 13
- 3.3 Relaciones de la unidad familiar 17
- 3.4 Resumen de información del cliente 19

Capítulo 4. Captura de datos relacionados 21

- 4.1 Introducción 21
- 4.2 Captura de datos compuestos 21
- 4.3 Visualización de datos compuestos en un resumen 22
- 4.4 Captura de datos asociados 22
- 4.5 Visualización de datos asociados en un resumen 23
- 4.6 Supresión de datos asociados 24

Capítulo 5. Formas eficientes de captura de datos 27

- 5.1 Introducción 27
- 5.2 Preguntas de lista 27
 - 5.2.1 Selección única 28
- 5.3 Preguntas de tabla de códigos 28
- 5.4 Elementos condicionales 30
 - 5.4.1 Secciones condicionales. 30
 - 5.4.2 Páginas condicionales 31
 - 5.4.3 Clústeres condicionales. 31
- 5.5 Matrices de preguntas 33
- 5.6 Navegación de Fast Path. 34
 - 5.6.1 Pregunta de lista que controla un bucle 34

- 5.6.2 Criterios de elegibilidad 35
- 5.6.3 Condiciones de Fast Path 36
- 5.6.4 Condición en bucle Fast Path. 38
- 5.7 Supresión implícita 38

Capítulo 6. Otras consideraciones sobre el desarrollo de script 39

- 6.1 Introducción 39
- 6.2 Visualización de datos como de sólo lectura 39
- 6.3 Invocación de funciones externas utilizando expresiones 40
- 6.4 Reutilización de scripts 45
- 6.5 Control de origen y versiones 46

Capítulo 7. Integración de IEG en una aplicación Cúram 47

- 7.1 Introducción 47
- 7.2 Creación de una ejecución de script 47
- 7.3 Especificación de un URL de redirección 47
- 7.4 Ejecución del reproductor IEG en una pestaña 47
- 7.5 Ejecución del reproductor IEG en un diálogo modal 50
 - 7.5.1 Apertura del reproductor IEG en un diálogo modal 50
 - 7.5.2 Salida de una ejecución de script en un diálogo modal 51
- 7.6 Limpieza de datos de aplicación 52
- 7.7 Reanudación de scripts ejecutados 53

Capítulo 8. Gestión de datos capturados 55

- 8.1 Introducción 55
- 8.2 Recuperación de datos capturados 55
- 8.3 Llenado previo de scripts con datos capturados 55

Capítulo 9. Utilización del almacén de recursos 59

- 9.1 Introducción 59
- 9.2 Listado de todos los recursos 59
- 9.3 Carga de un nuevo recurso 59
- 9.4 Eliminación de un recurso existente 60
- 9.5 Actualización de un recurso existente 60
- 9.6 Descarga de un recurso existente 60
- 9.7 Adición de imágenes 60
- 9.8 Cambio de texto estático. 60
- 9.9 Cambio de la codificación de archivo predeterminada 61

Capítulo 10. Utilización de IBM Rational AppScan para explorar IEG 63

- 10.1 Introducción 63
- 10.2 Preparación. 63
- 10.3 Páginas de relaciones 63

10.4 Configuración de la exploración	63	10.11 Completo	65
10.5 Política de pruebas	64	10.12 Ejecución de la exploración	65
10.6 Opciones de exploración	64	Avisos	67
10.7 Comunicaciones y proxy	64	Marcas registradas	69
10.8 Opciones de prueba	64		
10.9 Operaciones de varios pasos	64		
10.10 Excluir vías de acceso y archivos	65		

Figuras

1. Esquema inicial	6	34. Atributo de persona adicional	32
2. Entidad de persona	6	35. Clúster condicional estático	32
3. Esquema básico	7	36. Clúster dinámicamente condicional	32
4. Script nuevo	7	37. Atributo de abuso de sustancias.	33
5. Nueva sección	8	38. Ejemplo de código de matriz de pregunta	34
6. Clústeres, preguntas y texto de visualización	9	39. Ejemplo de código de pregunta de lista de Fast Path que controla un bucle	35
7. Página de resumen	10	40. Código de ejemplo de pregunta de lista de Fast Path con criterios de elegibilidad que controlan un bucle	36
8. Obtención del tamaño de la unidad familiar	14	41. Ejemplo de código de condiciones de Fast Path	37
9. Utilización del bucle "for" para recopilar miembros de la unidad familiar.	15	42. Ejemplo de condición en código de bucle Fast Path	38
10. Utilización del bucle "while" para recopilar miembros de la unidad familiar.	16	43. Establecimiento del distintivo de sólo lectura en una ejecución de script.	40
11. Utilización del bucle "while" para recopilar miembros de la unidad familiar.	17	44. Atributos de persona adicionales en el esquema de DS	41
12. Entidad de relación en un esquema de almacén de datos	18	45. Preguntas de estado y código postal en la definición de script	41
13. Página de relaciones	18	46. Función personalizada para validar el código postal	42
14. Bucle "for" para recopilar información de miembros de la unidad familiar.	19	47. Metadatos de función personalizada	42
15. Lista de personas	19	48. Validación de código postal en la definición de script	43
16. Lista resumen de relaciones	19	49. Expresión de validación alternativa	43
17. Esquema de padre/hijo	21	50. Función personalizada para llenar el estado	44
18. Creación de entidades anidadas.	21	51. Metadatos de función personalizada	44
19. Visualización de entidades anidadas en páginas de resumen.	22	52. Llamada para llenar el estado en la definición de script	44
20. Esquema de entidad asociada	23	53. Subscript que contiene páginas	45
21. Creación de relaciones de asociación	23	54. Inclusión de un subscript en un script	45
22. Página de resumen de asociación de entidades	24	55. Creación de una ejecución de script	47
23. Esquema de supresiones en cascada	25	56. Script con finish-page y quit-page definidos	47
24. Esquema Has Income Person.	27	57. UIM de resolución para abrir el reproductor IEG	49
25. Pregunta de lista	27	58. Supresión de la entidad raíz	53
26. Esquema de persona cuidador principal	28	59. Obtención de entidad raíz.	55
27. Single-select List Question	28	60. Fragmento de código que llena el DS	56
28. Tabla de códigos y atributo de estado	29	61. Creación de una ejecución de script	56
29. Pregunta de tabla de códigos de estado	29	62. Inicio del reproductor IEG	57
30. Entidad de estado	30		
31. Pregunta de tabla de códigos de selección múltiple.	30		
32. Atributo visible de una sección	31		
33. Sección condicional	31		

Tablas

1. Datos de cliente a capturar	5
--	---

Capítulo 1. Introducción

1.1 Público al que va dirigida

Esta publicación está destinada a los autores noveles en Intelligent Evidence Gathering (IEG - Recopilación de pruebas inteligente) que desean utilizar las características para capturar datos de forma inteligente como parte de una aplicación interna o externa. Técnicamente, estos datos pueden ser los que desee y se pueden utilizar para cualquier propósito, pero normalmente los datos en cuestión son datos relacionados de cliente y son necesarios como parte de una aplicación para un programa o para determinar la elegibilidad potencial. Toda la información de este tipo se proporciona bajo la cabecera general de pruebas en Cúram. Dado su estilo didáctico, esta guía hace referencia a usted, el diseñador de script, directamente.

1.2 Finalidad

La finalidad de esta guía es proporcionar a los autores de script la información esencial sobre cómo definir y mantener scripts IEG y los esquemas de almacén de datos (DS) asociados para utilizarlos en aplicaciones internas o externas.

IEG es una tecnología que se proporciona como parte de la suite de aplicaciones de Cúram, que permite a los clientes crear scripts dinámicos para recopilar datos de muchas maneras diferentes. Existen sin embargo algunas cuestiones a tener en cuenta al crear un script IEG y esquema de DS. En esta guía se describirán algunas de esas cuestiones a tener en cuenta, así como información relacionada con el mantenimiento de scripts.

1.3 Lectura adicional

Existen algunos otros documentos que se deben leer antes de crear y liberar un script IEG. En primer lugar, la publicación Cúram Development Compliancy Guide describe las restricciones que se aplican cuando se desarrollan aplicaciones utilizando IEG que necesitan conocerse antes de iniciar cualquier implementación. El otro documento que merece la pena leer es *Authoring Scripts using Intelligent Evidence Gathering (IEG)*. Este documento puede utilizarse como guía de referencia y contiene información detallada sobre todas las características disponibles en IEG e instrucciones sobre cómo utilizar estas funciones. La guía *Creating Datastore Schemas* explica cómo se crean y se mantienen los esquemas de DS para utilizarse con IEG.

Capítulo 2. Cómo empezar

2.1 Introducción

En este capítulo se explicarán los principios básicos de IEG y su dependencia del almacén de datos (DS) y del almacén de recursos (RS). Este capítulo le guiará durante la creación de un script IEG sencillo para recopilar información sobre un cliente.

2.2 Acerca de IEG

IEG es una alternativa eficaz a los procesos de recopilación de información tradicionales. Con IEG, la información se recopila de forma interactiva visualizando un script de preguntas a las que un usuario puede proporcionar respuestas. Las preguntas sólo se visualizan si son coherentes con las respuestas anteriores del usuario para que el usuario sólo tenga que proporcionar respuestas relevantes para sus necesidades y situación. Esto crea un entorno de uso agradable para el usuario que se puede implementar de forma efectiva para un rango de procesos que incluyen admisión de información de cliente, selección de valoración de prestaciones, valoración de elegibilidad en línea, etc.

A diferencia de los procesos de recopilación de información tradicionales, IEG reduce el trabajo administrativo de la organización creando el potencial de varias rutas a través del mismo script de preguntas. Esto elimina la necesidad de desarrollar muchos scripts para recopilar información de diferentes tipos de usuarios.

Una ventaja adicional de IEG es la flexibilidad de su implementación y el rango de los usuarios potenciales. El entorno de ejecución IEG se puede configurar para el acceso desde cualquier página UIM. Esto significa que se puede acceder directamente a IEG desde una aplicación de la organización o de forma remota mediante un usuario en línea.

Los dos componentes principales de IEG son el motor y el reproductor. Los scripts IEG se definen en XML y el motor interpreta las definiciones de script en el tiempo de ejecución y evalúa las respuestas suministradas por el usuario para determinar el flujo de ejecución. El motor determina qué páginas deben mostrarse al usuario y cuántas veces deben visualizarse. El reproductor presenta las páginas, las preguntas y otros controles al usuario. IEG también se basa en otros elementos de la suite de aplicaciones de Cúram, como el almacén de datos (DS) y el almacén de recursos (RS).

2.2.1 Almacén de datos (DS)

El propio IEG no conserva directamente los datos proporcionados por un usuario durante la ejecución de script. Esta tarea se delega al almacén de datos (DS). El DS es una base de datos de configurable. De mismo modo que las preguntas y las páginas de preguntas que se deben visualizar al usuario las determina un script IEG, los datos que se pueden almacenar en el DS los determina dinámicamente un esquema XML. El esquema describe la estructura de la información que desea almacenar y las relaciones entre los datos. Los datos se almacenan en el DS en formato XML y se ajustan al lenguaje de definición de esquemas XML W3C. Se pueden encontrar más detalles sobre el DS y cómo funciona en la guía de creación de esquemas de almacén de datos (Creating Datastore Schemas).

Un script IEG y un esquema de DS están muy estrechamente enlazados. Un script IEG se define con referencias a los elementos contenidos en un esquema y, por esa razón, se debe proporcionar un esquema al editar un script. El mismo esquema también se necesita cuando se ejecuta un script. Los esquemas se pueden reutilizar para editar y ejecutar varios scripts a fin de que se puedan utilizar las mismas estructuras de datos en circunstancias diferentes.

2.2.2 Almacén de recursos (RS)

Un script IEG puede contener referencias a imágenes que se visualizarán al usuario cuando se ejecute un script, por ejemplo, iconos que representan secciones y páginas de preguntas. Las imágenes se almacenan en el almacén de recursos (RS). Un script IEG también contiene diversos elementos de texto diferentes, por ejemplo cabeceras de página, etiquetas de pregunta y texto de ayuda. IEG le permite entrar todo el texto del script para la configuración regional predeterminada directamente en la definición de script.

Cuando se carga un script IEG en el sistema a través de las pantallas de administración IEG, todo el texto contenido en él se extrae automáticamente en un archivo de propiedades denominado adecuadamente para el script. Estos archivos de propiedades también se almacenan en el RS. Los archivos de propiedades se almacenan sin ninguna configuración regional asociada a ellos (para que actúen como propiedades de emergencia si no existen propiedades para la configuración regional en la que se está ejecutando). El RS permite cargar archivos de propiedades para varias configuraciones regionales haciendo que la localización de scripts sea una tarea sencilla. Durante el tiempo de ejecución, los archivos de propiedades se recuperan para el entorno local adecuado y se presentan al usuario en el reproductor IEG.

2.2.3 Estructura de script

En su forma más simple, un script IEG consta de páginas que incluyen preguntas que se plantean a los usuarios de IEG. La estructura del script IEG es una agrupación lógica de estas páginas, de modo que las respuestas a las preguntas se pueden capturar de forma eficaz. Las secuencias de páginas se pueden agrupar en secciones lógicas. La finalidad de estas secciones es proporcionar a los usuarios una vista de nivel superior de la clase de información capturada por el script IEG.

Además de incluir un número variable de páginas, cada sección debe contener una página de resumen. Esta página se utiliza para proporcionar comentarios al usuario sobre la información especificada en las páginas de una sección. Normalmente las páginas de resumen contienen clústeres y listas que visualizan versiones de sólo lectura de las respuestas a las preguntas formuladas. La página de resumen siempre será la última página visualizada dentro de una sección y también se visualizará siempre que un usuario pulse en el enlace para esa sección en la barra lateral del reproductor IEG.

Para resumir, los scripts IEG constan de una jerarquía de elementos estructurados de forma parecida a lo siguiente:

- Script
 - Sección
 - Página
 - Clúster
 - Pregunta
 - Página de resumen

2.3 Evaluación del uso de IEG

Hay algunas preguntas clave que se deben formular al evaluar el uso de IEG en cualquier aplicación:

- ¿Qué información se está capturando?
- ¿Cuál es el origen de esa información?
- ¿Cómo se debe utilizar esta información?
- ¿Cuánto tiempo estará activa esta información en la aplicación?

Muchos de los usos actuales de IEG surgen de la necesidad de soportar una aplicación para productos y servicios ofrecidos por agencias tanto externa como internamente. La información capturada es generalmente información relacionada con el cliente, como los detalles personales del cliente, los detalles de su familia o unidad familiar y los detalles de sus necesidades.

Normalmente las agencias ya tienen datos acerca de un cliente; por lo tanto, pueden obtener la información de otro sistema utilizando algunas partes clave de la información como un número de la seguridad social. Esto les permite verificar la información de cliente que se está entrando o recuperarla para ayudar con la aplicación.

Algunas aplicaciones son complejas y requieren información de muchos orígenes. Los clientes pueden tener que entrar información que no está a mano. Por ejemplo, es posible que la información necesaria la retenga el empleador. Es posible que necesiten poder almacenar lo que han entrado y volver a la aplicación más adelante una vez que tienen todos los datos necesarios.

Los clientes pueden estar expuestos a aplicaciones de exploración simples que les informen de sus derechos bajo la legislación actual o nueva. Normalmente esta información no es fiable y los datos temporales se deben eliminar del sistema después de que el cliente haya finalizado la sesión o dentro de un periodo de tiempo establecido.

Estos requisitos controlan el uso de IEG y proporcionan información importante sobre el uso de los datos durante su tiempo de vida.

Por lo tanto, vamos a empezar con los aspectos básicos: queremos capturar y almacenar información sobre un cliente.

2.4 Aspectos básicos

2.4.1 Crear un esquema

El primer paso en la captura de datos acerca de un cliente es crear un esquema DS. Esta sección proporciona un ejemplo de cómo crear un esquema básico que defina la captura de algunos datos de cliente generales.

El DS almacena los datos recopilados de los usuarios durante la exploración en línea y la admisión de solicitudes. El contenido del DS se puede definir dinámicamente mediante una definición de esquema. Los requisitos para capturar y almacenar datos acerca de un cliente pueden ser complejos pero con el diseño de esquema apropiado, estos datos se pueden gestionar de forma eficiente a lo largo del tiempo de vida.

Para este ejemplo, el requisito es capturar lo siguiente:

Tabla 1. Datos de cliente a capturar

Atributos	Tipo
Nombre	Serie
Segundo nombre	Serie
Apellido/Nombre de familia	Serie
Sexo	Hombre/Mujer
Fecha de nacimiento	Fecha

Existe un conjunto mínimo de definiciones necesarias en un esquema. Para un esquema que se debe utilizar en IEG, se necesita lo siguiente:

- Inclusión de dominios base
- Inclusión de dominios IEG
- Una entidad raíz

Para obtener más información sobre el conjunto mínimo de definiciones necesarias, consulte la guía Creación de esquemas de almacén de datos.

El esquema tendrá un aspecto similar al del siguiente antes de añadir contenido nuevo como la entidad de persona descrita más arriba:

```
<xsd:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:d="http://www.curamsoftware.com/BaseDomains">
  <xsd:import namespace="http://www.curamsoftware.com/BaseDomains"/>
  <xsd:include schemaLocation="IEGDomains"/>
  <xsd:element name="Application">
    <xsd:complexType>
      <xsd:sequence minOccurs="0">
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Figura 1. Esquema inicial

El contenido del esquema indica que se trata de un esquema XML que importa el esquema BaseDomains e incluye el esquema IEGDomains. El primer elemento denominado Application es la entidad raíz para el esquema. IEG necesita que la entidad raíz se denomine siempre Application.

El esquema IEGDomains contiene los dominios necesarios para definir los atributos de las entidades que se van a utilizar con IEG. Los tipos de los atributos se deben derivar de los dominios IEG, en lugar de los dominios base. Una entidad de persona puede definirse para representar un cliente como se indica a continuación:

```
<xsd:element name="Person">
  <xsd:complexType>
    <xsd:attribute name="firstName" type="IEG_STRING"/>
    <xsd:attribute name="middleName" type="IEG_STRING"/>
    <xsd:attribute name="lastName" type="IEG_STRING"/>
    <xsd:attribute name="gender" type="IEG_GENDER"/>
    <xsd:attribute name="dateOfBirth" type="IEG_DATE"/>
  </xsd:complexType>
</xsd:element>
```

Figura 2. Entidad de persona

Hay un par de cuestiones a tener en cuenta acerca de la adición anterior para una entidad como persona:

- Al igual que las tablas de base de datos relacional, se necesita un campo de ID y se define una clave para esta tabla utilizando este ID exclusivo.
- La entidad de persona se añade como una entidad hijo de la entidad raíz.

El esquema para capturar información básica acerca de una persona puede definirse como se indica a continuación:


```

<xsd:element name="Application">
  <xsd:complexType>
    <xsd:sequence minOccurs="0">
      <xsd:element ref="Person" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Person">
  <xsd:complexType>
    <xsd:attribute name="personID" type="d:SVR_KEY"/>
    <xsd:attribute name="firstName" type="IEG_STRING"/>
    <xsd:attribute name="middleName" type="IEG_STRING"/>
    <xsd:attribute name="lastName" type="IEG_STRING"/>
    <xsd:attribute name="gender" type="IEG_GENDER"/>
    <xsd:attribute name="dateOfBirth" type="IEG_DATE"/>
  </xsd:complexType>
  <xsd:key name="Person_Key">
    <xsd:selector xpath="./Person"/>
    <xsd:field xpath="@personID"/>
  </xsd:key>
</xsd:element>

```

Figura 3. Esquema básico

Una vez que se ha definido el esquema, puede crear un script para utilizar el esquema.

2.4.2 Crear un script

IEG le permite crear scripts dinámicos para recopilar datos. Los scripts IEG pueden contener secciones, páginas de preguntas, preguntas y lógica condicional que le permite decidir qué información se debe capturar, qué páginas se deben visualizar y cuántas veces se visualizan.

Lea la guía *Authoring Scripts using Intelligent Evidence Gathering (IEG)* para obtener detalles sobre cómo definir cada elemento de un script IEG.

Para los requisitos anteriores, donde es necesario capturar información acerca de una persona, debe definir el script y decidir cómo se organizan las páginas para capturar la información.

Se puede crear un nuevo script en la aplicación de administración y se puede utilizar el editor para añadir elementos a este script. El contenido de un script recién creado será similar a lo siguiente:

```

<?xml version="1.0" encoding="UTF-8"?>
<ieg-script xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ieg-schema.xsd">
  <identifier id="WorkingWithIEG" scriptversionnumber="V1"
    type="Intake" />
</ieg-script>

```

Figura 4. Script nuevo

Los ID, Type y Version proporcionados al crear el script se combinan para crear un identificador de script para identificar de forma exclusiva la definición de script.

Una vez que se ha creado un script nuevo, se pueden añadir al script elementos como secciones, páginas de preguntas y páginas de resumen. Los ejemplos de las dos secciones siguientes muestran cómo añadir una sección y una página de preguntas a un script, así como la forma de añadir una página de resumen que vuelva a mostrar información al usuario. Las páginas de resumen permiten al usuario confirmar que los datos que ha entrado son correctos antes de continuar y también pueden proporcionar al usuario la posibilidad de modificar los datos.

2.4.2.1 Adición de una sección y una página de preguntas a un script IEG

Es necesario añadir una sección y una página de preguntas. Se puede utilizar una sección para agrupar las páginas relacionadas a fin de permitir que el usuario fluya por las pantallas de una forma lógica. Las secciones también pueden ayudar a transmitir al usuario su progreso a través de un script. Tanto la sección como la página de preguntas pueden tener un título y la página de preguntas puede tener opcionalmente una descripción.

El ejemplo de código siguiente muestra una sección que contiene una página de preguntas, añadida a un script:

```
<?xml version="1.0" encoding="UTF-8"?>
<ieg-script xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ieg-schema.xsd">
  <identifier id="WorkingWithIEG" scriptversionnumber="V1"
    type="Intake" />
  <section>
    <title id="AboutYouSection.Title">
      <![CDATA[Acerca de usted]]>
    </title>
    <question-page id="AboutYouPage" entity="Person">
      <title id="PrimaryPersonPage.Title">
        <![CDATA[Acerca de usted]]>
      </title>
      <description id="PrimaryPersonPage.Description">
        <![CDATA[Entre información sobre usted]]>
      </description>
    </question-page>
  </section>
</ieg-script>
```

Figura 5. Nueva sección

La página de preguntas requiere las preguntas adecuadas para capturar los datos. Los datos que se deben almacenar en el DS tienen que estar asociados con un atributo de una entidad del esquema DS para utilizarse con este script. Si todas las preguntas de una página se refieren a la misma entidad, la página puede correlacionarse con ese tipo de entidad. En el ejemplo anterior, la página se correlaciona con la entidad de persona.

Para añadir preguntas a una página, se necesita un clúster. Los clústeres ayudan a controlar el diseño de las preguntas en la página. Una página puede contener muchos clústeres para permitirle agrupar preguntas de forma lógica en la página. Los clústeres también pueden contener un título y una descripción.

En nuestro ejemplo siguiente, hay dos clústeres, uno sólo para visualizar texto informativo al usuario y otro para contener las preguntas para detalles personales. Se pueden añadir preguntas y texto de visualización a cada clúster. Se debe proporcionar un ID a las preguntas que debe corresponder al de los atributos del tipo de entidad con el que está correlacionada la página. Si se debe proporcionar una respuesta a una pregunta, se puede establecer el indicador obligatorio de la pregunta en true. El siguiente fragmento de script contiene las preguntas para capturar los datos necesarios descritos en nuestro ejemplo.

```

<question-page ...
  <cluster>
    <display-text id="RequiredFields.Text">
      <![CDATA[<span style="color: orange;">
        * indica un campo necesario</span>]]>
    </display-text>
  </cluster>
  <cluster>
    <title id="DetailsCluster.Title">
      <![CDATA[Detalles personales]]>
    </title>
    <description id="DetailsCluster.Description">
      <![CDATA[Entre los detalles aquí]]>
    </description>
    <question id="firstName" mandatory="true">
      <label id="FirstName.Label">
        <![CDATA[Nombre:]]>
      </label>
    </question>
    <question id="middleName">
      <label id="MiddleName.Label">
        <![CDATA[Segundo nombre:]]>
      </label>
    </question>
    <question id="lastName">
      <label id="lastName.Label">
        <![CDATA[Apellidos:]]>
      </label>
    </question>
    <question id="gender" mandatory="true">
      <label id="Gender.Label">
        <![CDATA[Sexo:]]>
      </label>
    </question>
    <question id="dateOfBirth" mandatory="true">
      <label id="DateOfBirth.Label">
        <![CDATA[Fecha de nacimiento:]]>
      </label>
    </question>
  </cluster>
</question-page>

```

Figura 6. Clústeres, preguntas y texto de visualización

Tenga en cuenta que existen más propiedades de scripts, secciones, páginas de preguntas, clústeres, preguntas y textos de visualización de las que se describen aquí. Algunas de estas propiedades que se cubren en la guía *Authoring Scripts using Intelligent Evidence Gathering (IEG)*, se describen más adelante en esta guía.

2.4.3 Adición de una página de resumen a un script IEG

El paso final de este ejemplo básico es mostrar un resumen de la información capturada. Generalmente cada sección tendrá una página de resumen. Una página de resumen se utiliza para volver a mostrar los datos más importantes al usuario para que éste verifique si los los datos se han capturado o calculado correctamente. Una página de resumen puede mostrar datos capturados en varias páginas de preguntas. Una página de resumen no tiene que contener toda la información capturada en la sección, ya que esto podría ser muy grande haciéndose entonces menos útil.

Es evidente que si los datos visualizados en una página de resumen son incorrectos, el usuario probablemente querrá modificarlos. Los usuarios pueden navegar hacia atrás en la ejecución de script pulsando el botón Atrás en el reproductor IEG hasta que lleguen a la página donde se han entrado los datos, actualizar los datos y, a continuación, seguir adelante de nuevo por el script. De forma alternativa, puede añadir enlaces de edición a los clústeres de la página de resumen. Cuando el usuario pulsa en un enlace de edición de una página de resumen, la página de pregunta especificada en el enlace de edición

se muestra al usuario en el reproductor IEG. Entonces el usuario puede cambiar los datos y dependiendo de si se hace referencia en cualquier otro lugar del script a los datos cambiados, la página de resumen se vuelve a visualizar cuando el usuario pulsa el botón Siguiente en el reproductor IEG.

La página de resumen en este caso será muy sencilla y similar a la página de preguntas previamente añadida. Y de forma similar a una página de preguntas, si todos los atributos a los que se hace referencia en la página están relacionados con la misma entidad, la página de resumen se puede correlacionar con ese tipo de entidad, como se indica a continuación:

```
<section>
...
<summary-page id="AboutYouSummary" entity="Person">
  <title id="AboutYouSummary.Title">
    <![CDATA[Información acerca de usted]]>
  </title>
  <description id="AboutYouSummary.Description">
    <![CDATA
      [Aquí está la información que ha entrado sobre usted]]>
  </description>
  <cluster>
    <title id="DetailsCluster.Title">
      <![CDATA[Detalles de la persona]]>
    </title>
    <description id="DetailsCluster.Description">
      <![CDATA[Entre los detalles de esta persona aquí]]>
    </description>
    <edit-link start-page="AboutYouPage" />
    <question id="firstName">
      <label id="FirstName.Label">
        <![CDATA[Nombre:]]>
      </label>
    </question>
    <question id="middleName">
      <label id="MiddleName.Label">
        <![CDATA[Segundo nombre:]]>
      </label>
    </question>
    <question id="lastName">
      <label id="lastName.Label">
        <![CDATA[APELLIDOS:]]>
      </label>
    </question>
    <question id="gender">
      <label id="Gender.Label">
        <![CDATA[Sexo:]]>
      </label>
    </question>
    <question id="dateOfBirth">
      <label id="DateOfBirth.Label">
        <![CDATA[Fecha de nacimiento:]]>
      </label>
    </question>
  </cluster>
</summary-page>
</section>
```

Figura 7. Página de resumen

Este script y esquema básicos para capturar información sobre una persona y visualizar una página de resumen ahora está completo y se puede ejecutar.

2.4.4 Ejecutar un script

Para ejecutar un script IEG se debe cargar en el sistema la definición de script y la definición de esquema asociado. Hay un número de maneras en las que se puede hacer esto que se cubren más adelante en esta

guía. El modo más sencillo para cargar las definiciones es a través de las pantallas de administración de la sección de recopilación de pruebas inteligente del espacio de trabajo de administración.

Para obtener acceso a las pantallas de administración de IEG, necesitará iniciar la sesión como usuario administrativo. Una vez iniciada la sesión, verá una sección en el panel de atajos denominada Recopilación de pruebas inteligente y, cuando pulse en ella, verá un menú para 'IEG' que contiene un enlace denominado 'Scripts'. Si pulsa en él, verá una página que contiene una lista de los scripts IEG actualmente en el sistema y varios enlaces para que pueda realizar actividades en estos scripts.

En la parte superior de la página 'Scripts' hay un enlace 'Importar' que le permite cargar o importar una nueva definición de script IEG.

Del mismo modo, si pulsa en el enlace 'Esquemas de almacén de datos' del menú para 'IEG', verá una página que contiene una lista de los esquemas de DS actualmente en el sistema. En la parte superior de la página 'Esquemas de almacén de datos', hay también un enlace 'Importar' que le permite cargar o importar una nueva definición de esquema.

Para su comodidad, IEG proporciona un tipo de arnés de prueba que permite que los scripts IEG se prueben sin tener que integrarlos en la aplicación Cúram. El arnés de prueba tiene algunas limitaciones, pero permite probar la mayoría de scripts tan pronto como se cargan en el sistema. Los scripts IEG pueden ejecutarse en una pestaña o en una ventana modal a través de las pantallas de administración.

Un script se puede ejecutar utilizando las opciones 'Ejecutar' o 'Ejecutar en modal' para el script de la página 'Scripts'. Como no hay ninguna asociación explícita entre un script IEG y un esquema de DS, cuando se selecciona la opción de ejecutar un script, se le pedirá que seleccione un esquema en un desplegable con el que deberá ejecutar el script. Al pulsar en el botón 'Ejecutar script', el reproductor IEG se iniciará y se le mostrará la primera página del script.

2.4.4.1 Validación de un script

Cuando un script se ejecuta a través de las pantallas de administración de esta forma, el script se valida antes de ejecutarse. También puede elegir la opción 'Validar' para el script en la página 'Scripts'. Todos los scripts deben validarse antes de ejecutarlos. Si el script no logra validarse, se visualizará una lista de errores de validación. Los errores de validación deben solucionarse antes de que el script se pueda ejecutar desde la página 'Scripts'.

Rellene algunos datos de ejemplo en la primera página del script y seleccione el botón Siguiente. Ahora estos mismos datos de ejemplo se deben visualizar en la página de resumen. Las respuestas no son modificables pero se proporciona un enlace de edición para volver a la página donde se han entrado esos datos.

Tenga en cuenta que al pulsar el botón Siguiente en el reproductor IEG en la página de resumen del script que se ha implementado en este ejemplo se visualizará un error. Esto se debe a que no se han definido todas las propiedades del script. Las propiedades necesarias se tratarán más adelante en esta guía.

Capítulo 3. Captura de información de cliente

3.1 Introducción

En el capítulo anterior se describía un ejemplo básico de cómo se puede utilizar IEG para capturar datos para un cliente. Algunos formularios de solicitud para prestaciones y servicios pueden ser complejos y la información necesaria sobre los solicitantes puede ser muy detallada.

Empezaremos a construir en el ejemplo inicial del capítulo anterior, considerando una unidad familiar, donde hemos capturado algunos datos iniciales acerca de un miembro principal primario y ahora deseamos añadir detalles para los demás miembros de la unidad familiar.

3.2 Familias y unidades familiares

Tenemos actualmente tienen un script sencillo, relacionado con una persona. Normalmente las solicitudes necesitan más información sobre las circunstancias del cliente, empezando por su situación de vida.

En general, se solicita información sobre la persona principal y a esto le sigue una pregunta sencilla que permitirá que el cliente salte a otra área de la solicitud. Por ejemplo, después de entrar detalles personales, se le pregunta al cliente "¿Vive solo?". Si la respuesta es afirmativa, se puede tratar a la persona como un solo individuo que no vive en una unidad familiar compuesta por una familia u otros individuos. La mayoría de los clientes desean pasar por el proceso de solicitud lo más rápidamente posible, por lo tanto preguntas como estas proporcionan una buena forma de pasar a partes más relevantes de la solicitud.

Si el cliente vive con otras personas, es posible que sea necesario realizar preguntas sobre cada persona. Se utilizan bucles para capturar información de cada persona y, en función de la manera en que desee presentar estas preguntas, el autor del script puede elegir estos tipos de bucle: "for", "while" y "for-each".

IEG también presenta una pestaña de personas que permite al cliente ver con quién están relacionadas estas preguntas mientras se entran los datos. Esto aparecerá automáticamente para una entidad de persona en el almacén de datos. Cada persona se representará mediante un icono (basado en el sexo y edad) y un nombre. La persona actual estará resaltada.

Veamos un escenario para manejar datos de familia/unidad familiar como una extensión de los requisitos del ejemplo básico. Aquí, se le pregunta al cliente cuántas personas hay en la unidad familiar, incluido el cliente. Es necesario añadir algunas páginas de preguntas nuevas para capturar esta información.

En la primera página de preguntas se pregunta sobre la situación que viven. En este ejemplo, sólo se puede formular una pregunta: ¿Cuántas personas hay en la familia (excluyéndose usted)?

```

<question-page id="HouseholdPage" progress="10">
  <title id="LoopControlPage.Title">
    <![CDATA[Detalles de la unidad familiar]]>
  </title>
  <description id="LoopControlPage.Description">
    <![CDATA[Proporcione información sobre su
    unidad familiar]]>
  </description>
  <icon image="sample_title_household" />
  <cluster>
    <title id="DetailsCluster.Title">
      <![CDATA[Detalles]]>
    </title>
    <question id="numPeople" control-question="true"
      control-question-type="IEG_INT32"
      mandatory="true">
      <label id="NumPeople.Label">
        <![CDATA[¿Cuántas personas hay en su
        unidad familiar?]]>
      </label>
    </question>
  </cluster>
</question-page>

```

Figura 8. Obtención del tamaño de la unidad familiar

Esta es una pregunta de control, es decir, una pregunta utilizada para controlar el tamaño de un bucle y no para recopilar datos. Las preguntas de control no se almacenan en el esquema de almacén de datos. Se utilizará en la expresión de bucle para el bucle "for" en la página de preguntas siguiente.

La página de preguntas de los miembros de la familia está dentro de un bucle "for" que se repetirá para todos los miembros de la familia.


```

<loop loop-type="for" loop-expression="numPeople"
  entity="Person" criteria="isPrimary==false">
  <question-page id="PersonDetailsPage"
    show-person-tabs="true"
    progress="20">
    <title id="PersonDetailsPage.Title">
      <![CDATA[Detalles de miembros de la unidad familiar]]>
    </title>
    <description id="PersonDetailsPage.Description">
      <![CDATA[Especifique los detalles de la
        siguiente persona de su unidad familiar]]>
    </description>
    <icon image="sample_title_household" />
    <cluster>
      <title id="DetailsCluster.Title">
        <![CDATA[Detalles de la persona]]>
      </title>
      <description id="DetailsCluster.Description">
        <![CDATA[Especifique los detalles de esta persona
          más abajo]]>
      </description>
      <question id="firstName" mandatory="true">
        <label id="FirstName.Label">
          <![CDATA[Nombre:]]>
        </label>
      </question>
      <question id="lastName">
        <label id="lastName.Label">
          <![CDATA[Apellidos:]]>
        </label>
      </question>
      <question id="gender" mandatory="true">
        <label id="Gender.Label">
          <![CDATA[Sexo:]]>
        </label>
      </question>
    </cluster>
  </question-page>
</loop>

```

Figura 9. Utilización del bucle "for" para recopilar miembros de la unidad familiar

Lo anterior es un ejemplo de cómo el cliente entra el número de miembros de la familia. Pero la pregunta podía haberse formulado de una forma diferente, por ejemplo: "¿Vive con su familia?" En este caso, se puede utilizar un elemento de condición del script para comprobar el valor de esa pregunta. Esto visualizará la página de miembro de la familia si efectivamente viven con su familia. En esta página de preguntas, se realiza una pregunta de control para determinar si desean capturar detalles de otro miembro de la familia.

Esta pregunta de control se utilizará en un bucle "while" en toda la página de preguntas de miembro de la familia, tal como se indica a continuación:

```

<question-page id="HouseholdPage" progress="10">
  <title id="LoopControlPage.Title">
    <![CDATA[Detalles de la unidad familiar]]>
  </title>
  <description id="LoopControlPage.Description">
    <![CDATA[Proporcione información sobre su
    unidad familiar]]>
  </description>
  <icon image="sample_title_household" />
  <cluster>
    <title id="DetailsCluster.Title">
      <![CDATA[Detalles]]>
    </title>
    <question id="livesWithFamily" control-question="true"
      control-question-type="IEG_BOOLEAN"
      mandatory="true">
      <label id="NumPeople.Label">
        <![CDATA[¿Vive con su familia?]]>
      </label>
    </question>
  </cluster>
</question-page>

```

Figura 10. Utilización del bucle "while" para recopilar miembros de la unidad familiar

Con este enfoque, la pregunta de control es un tipo booleano, porque se utiliza en una expresión de condición que indica si se debe entrar en el bucle while. Una vez que se ha entrado en él, el bucle se itera hasta que se han reunido detalles de todos los miembros de la unidad familiar, como se indica a continuación:

```

<condition expression="livesWithFamily==true">
  <loop loop-type="while" loop-expression="
    anotherMember==true"
    entity="Person">
    <question-page id="PersonDetailsPage"
      show-person-tabs="true"
      progress="20">
      <title id="PersonDetailsPage.Title">
        <![CDATA[Detalles de miembros de la unidad familiar]]>
      </title>
      <description id="PersonDetailsPage.Description">
        <![CDATA[Especifique los detalles de
          la siguiente persona de la unidad familiar]]>
      </description>
      <icon image="sample_title_household" />
      <cluster>
        <title id="DetailsCluster.Title">
          <![CDATA[Detalles de la persona]]>
        </title>
        <description id="DetailsCluster.Description">
          <![CDATA[Especifique los detalles de esta
            persona más abajo]]>
        </description>
        <question id="firstName" mandatory="true">
          <label id="FirstName.Label">
            <![CDATA[Nombre:]]>
          </label>
        </question>
        <question id="lastName">
          <label id="lastName.Label">
            <![CDATA[APELLIDOS:]]>
          </label>
        </question>
        <question id="gender" mandatory="true">
          <label id="Gender.Label">
            <![CDATA[Sexo:]]>
          </label>
        </question>
      </cluster>
      <cluster>
        <question id="anotherMember"
          control-question="true"
          control-question-type="IEG_BOOLEAN">
          <label id="AnotherMember.Label">
            <![CDATA[¿Hay otro
              miembro de la unidad familiar?]]>
          </label>
        </question>
      </cluster>
    </question-page>
  </loop>
</condition>

```

Figura 11. Utilización del bucle "while" para recopilar miembros de la unidad familiar

3.3 Relaciones de la unidad familiar

Al recopilar información sobre un grupo de personas de una unidad familiar, es posible que sea necesario determinar cómo están relacionadas esas personas entre sí. IEG proporciona un mecanismo para capturar relaciones mediante el uso de páginas de relaciones y una estructura de esquema de almacén de datos específica.

Se debe definir una entidad de relación en el esquema de almacén de datos, tomando el formato siguiente:

```
<xsd:element name="Person">
  <xsd:complexType>
    <xsd:sequence minOccurs="0">
      <xsd:element ref="Relationship" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    ...
  </xsd:element>
<xsd:element name="Relationship">
  <xsd:complexType>
    <xsd:attribute name="relationshipType"
      type="IEG_STRING"/>
    <xsd:attribute name="isNonParentPrimaryCaretaker"
      type="IEG_BOOLEAN" default="false"/>
    <xsd:attribute name="personID" type="D:SVR_KEY"/>
  </xsd:complexType>
</xsd:element>
```

Figura 12. Entidad de relación en un esquema de almacén de datos

Una página de relaciones para la unidad familiar se puede definir de la manera siguiente, a condición de que la entidad de relación sea un hijo de la entidad de persona:

```
<relationship-page id="RelationshipPage" show-person-tabs="true"
  progress="40">
  <title id="RelationshipPage.Title">
    <![CDATA[Relaciones de unidad familiar]]>
  </title>
  <description id="RelationshipPage.Description">
    <![CDATA[Entre las relaciones para %1s más abajo]]>
    <argument id="Person.firstName" />
  </description>
  <icon image="sample_title_household" />
  <question id="caretakerInd">
    <label id="CaretakerInd.Label">
      <![CDATA[¿Es una relación de tutor
        no parental?]]>
    </label>
  </question>
</relationship-page>
```

Figura 13. Página de relaciones

Sólo es necesario definir la página de relaciones una vez. Entonces IEG mostrará la página tantas veces como sea necesario para recopilar las relaciones de persona en persona. Esto equivale a una vez menos que el número de personas de la unidad familiar, porque las relaciones de la última persona se habrán recopilado en su totalidad a lo largo del proceso.

De forma predeterminada, el campo Tipo de relación se presenta como un desplegable, lleno con una tabla de códigos (configurable a través de la propiedad `relationship.type.domain.name`):

La página de relaciones mostrará una pestaña de personas en la parte superior que contiene la lista de miembros de la unidad familiar y la persona actual estará resaltada. Entonces se visualizará cada relación entre la persona actual y los demás miembros.

El indicador de tutor es la única pregunta que se puede añadir directamente a la página de relaciones. Las preguntas sobre otros atributos de una entidad de relación deben añadirse a los clústeres que se han añadido a la página de relaciones.

3.4 Resumen de información del cliente

Las listas se utilizan en las páginas de resumen para visualizar la información recopilada en bucles. La estructura de la lista debe reflejar la estructura del bucle o la jerarquía de los bucles que han recopilado los datos. Esto significa que la entidad y los criterios en la lista deben coincidir con la entidad y los criterios en el bucle. Por ejemplo, para registrar los miembros de la familia que se describe en 3.2, "Familias y unidades familiares", en la página 13, se ha utilizado un bucle:

```
<loop loop-type="for" loop-expression="numPeople"
  entity="Person" criteria="isPrimary==false">
  ...
</loop>
```

Figura 14. Bucle "for" para recopilar información de miembros de la unidad familiar

En la página de resumen de sección, la información recopilada en este bucle se visualiza en una lista. La lista, igual que el bucle, tiene 'Person' como entidad y 'isPrimary==falso' como criterios:

```
<list entity="Person" criteria="isPrimary==false">
  ...
</list>
```

Figura 15. Lista de personas

La información de relación recopilada utilizando una página de relaciones puede visualizarse en las páginas de resumen de las listas resumen de relaciones:

```
<relationship-summary-list>
  <title id="RelationshipSummaryList.Title">
    <![CDATA[Resumen de relaciones de persona]]>
  </title>
  <description id="PersonRelationshipSummaryList.Description">
    <![CDATA[Detalles de resumen de relaciones de persona]]>
  </description>
  <column id="caretakerInd">
    <title id="CaretakerInd.Title">
      <![CDATA[NPCR]]>
    </title>
  </column>
  <edit-link start-page="RelationshipPage" />
</relationship-summary-list>
```

Figura 16. Lista resumen de relaciones

Capítulo 4. Captura de datos relacionados

4.1 Introducción

Ahora que hemos capturado información acerca de los miembros de la unidad familiar, por ejemplo los detalles personales y las relaciones, puede que deseemos capturar datos relacionados. Esto puede conseguirse mediante la composición (el uso de entidades DS anidadas) o la asociación (el uso de entidades DS relacionadas no anidadas).

4.2 Captura de datos compuestos

Hemos visto anteriormente que es posible capturar relaciones en IEG. La combinación de la entidad de relación y la página de relaciones proporcionan un mecanismo adecuado para capturar las relaciones entre las personas de una unidad familiar. La relación entre las personas de una unidad familiar es sólo una forma de relación. IEG soporta otros tipos de relaciones. IEG y el DS permiten que las entidades se aniden creando una relación de padre e hijo. Esto puede verse en el ejemplo donde hay un requisito para capturar los ingresos de las personas de una unidad familiar. Una entidad Ingresos se define igual que cualquier otra entidad. Se anida en la entidad de Persona haciendo referencia a la misma en una secuencia, como se muestra en el siguiente fragmento de código de ejemplo:

```
<xsd:element name="Person">
  <xsd:complexType>
    <xsd:sequence minOccurs="0">
      <xsd:element ref="Income" minOccurs="0"
        maxOccurs="unbounded" />
    </xsd:sequence>
    ...
    <xsd:attribute name="hasIncome" type="IEG_BOOLEAN"
      default="false"/>
  </xsd:complexType>
  ...
</xsd:element>
<xsd:element name="Income">
  <xsd:complexType>
    <xsd:attribute name="type" type="IEG_STRING" />
    <xsd:attribute name="amount" type="IEG_MONEY" />
  </xsd:complexType>
</xsd:element>
```

Figura 17. Esquema de padre/hijo

A continuación, se puede recopilar información de ingresos para las personas de una unidad familiar poniendo en un bucle cada persona que tiene ingresos. Los criterios de bucle utilizarán una pregunta booleana "hasIncome" que se formulará al recopilar los detalles de cada persona. Una página dentro del bucle se puede correlacionar con la entidad de ingresos creando así la relación anidada, como se muestra a continuación:

```
<loop loop-type="for-each" entity="Person"
  criteria="hasIncome==true">
  <loop loop-type="while" loop-expression="hasMoreIncome"
    entity="Income">
    <question-page id="IncomePage" entity="Income"
    ...
```

Figura 18. Creación de entidades anidadas

4.3 Visualización de datos compuestos en un resumen

La información recopilada para entidades anidadas se pueden visualizar en una página de resumen utilizando una lista anidada. De forma similar a las listas normales, las listas anidadas deben coincidir con las entidades y los criterios utilizados en los bucles anidados que han capturado los datos.

```
<list entity="Person" show-icons="true" criteria="hasIncome==true">
  <title id="IncomeList.Title">
    <![CDATA[Ingresos]]>
  </title>
  <description id="IncomeList.Description">
    <![CDATA[Aquí están los detalles de ingresos que ha entrado para todas las
      personas de la unidad familiar]]>
  </description>
  <column id="firstName">
    <title id="FirstName.Title">
      <![CDATA[Nombre]]>
    </title>
  </column>
  <list entity="Income">
    <column id="type">
      <title id="IncomeType.Title">
        <![CDATA[Tipo de ingresos]]>
      </title>
    </column>
    <column id="amount">
      <title id="IncomeAmount.Title">
        <![CDATA[Importe de ingresos]]>
      </title>
    </column>
  </list>
</list>
```

Figura 19. Visualización de entidades anidadas en páginas de resumen

El fragmento de código de ejemplo anterior a una lista de resumen de ingresos se visualizará en el reproductor IEG como una lista normal con los ingresos agrupados por persona. También contendrá los enlaces Editar y Suprimir para cada uno de los ingresos y un enlace Añadir con un desplegable que lista todas las personas.

4.4 Captura de datos asociados

IEG permite que se creen relaciones de asociación entre entidades. Esto es útil porque una restricción se aplica a entidades anidadas y listas anidadas que sólo pueden estar anidadas en dos niveles. El uso de relaciones asociadas proporciona una alternativa efectiva a la anidación de entidades en tres niveles.

Por ejemplo, suponga que es necesario registrar información de empleo para las personas de una unidad familiar. Se puede recopilar información de empleo independientemente de la información de ingresos ya que pueden haber varios ingresos para un determinado empleo.

Una vez que se ha recopilado la información de ingresos y empleo y se han creado las entidades, se puede realizar la asociación entre las entidades. La asociación se realiza creando una entidad de "relación". La entidad de relación es normalmente "propiedad" de una de las entidades que participan en la relación y se representa como una secuencia igual que con otros tipos de relaciones.

La definición de una entidad de relación necesita poder identificar la entidad relacionada por lo tanto se debe definir una clave en la entidad relacionada. Para aplicar esto en el ejemplo de ingresos y empleo, el tipo de entidad Empleo tendrá una clave, se definirá un tipo de entidad EmploymentRelationship y la entidad de ingresos será propietaria de una secuencia de EmploymentRelationships, como se indica a continuación:


```

<xsd:element name="Employment">
  <xsd:complexType>
    <xsd:attribute name="employmentID" type="d:SVR_KEY" />
    <xsd:attribute name="employer" type="IEG_STRING" />
    <xsd:attribute name="employmentType" type="IEG_STRING" />
  </xsd:complexType>
  <xsd:key name="Employment_Key">
    <xsd:selector xpath="./Employment" />
    <xsd:field xpath="@employmentID" />
  </xsd:key>
</xsd:element>
<xsd:element name="Income">
  <xsd:complexType>
    <xsd:sequence minOccurs="0">
      <xsd:element ref="EmploymentRelationship" minOccurs="0"
        maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="type" type="IEG_STRING" />
    <xsd:attribute name="amount" type="IEG_MONEY" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="EmploymentRelationship">
  <xsd:complexType>
    <xsd:attribute name="employmentID" type="d:SVR_KEY" />
  </xsd:complexType>
</xsd:element>

```

Figura 20. Esquema de entidad asociada

Entonces la asociación se puede capturar en el script definiendo una lista de preguntas y especificando un atributo de entidad de enlace que hace referencia a la clave de la entidad relacionada. Siguiendo en el ejemplo, en una página correlacionada con la entidad de ingresos se puede definir una lista de preguntas especificando la clave de la EmploymentRelationship utilizada para identificar la entidad de empleo.

Las preguntas de lista son construcciones que permiten al usuario elegir en una lista de entidades. Para obtener más detalles, consulte 5.2, “Preguntas de lista”, en la página 27.

```

<question-page id="IncomePage" entity="Income" ...
  <cluster>
    <layout>
      <label-width>0</label-width>
    </layout>
    <list-question link-entity="EmploymentRelationship.employmentID"
      entity="Employment" single-select="true">
      <label id="SelectEmployer.Label">
        <![CDATA[Seleccionar empleador]]>
      </label>
      <item-label>
        <label-element attribute-id="employer" />
      </item-label>
    </list-question>
  </cluster>
</question-page>

```

Figura 21. Creación de relaciones de asociación

4.5 Visualización de datos asociados en un resumen

La asociación entre entidades se puede visualizar en una página de resumen añadiendo una columna a la lista de entidades de un tipo, para mostrar los detalles de la entidad relacionada. Es necesario especificar un atributo link-entity en esta columna para identificar la entidad relacionada.

El ejemplo siguiente muestra cómo, mientras se listan los ingresos para una persona en una página de resumen, se puede visualizar el nombre de empleador asociado para cada Ingreso:

```
<summary-page id="IncomeSummary"
...
<list entity="Person" criteria="hasIncome==true"
      show-icons="true">
  <title id="IncomeList.Title">Ingresos</title>
  <description id="IncomeList.Description">Aquí están los detalles
    de ingresos que ha entrado para todas las personas de la
    unidad familiar</description>
  <column id="firstName">
    <title id="FirstName.Title">Nombre</title>
  </column>
  <list entity="Income" show-icons="false">
    <column id="type">
      <title id="IncomeType.Title">Tipo de ingresos</title>
    </column>
    <column id="amount">
      <title id="IncomeAmount.Title">Importe de ingresos</title>
    </column>
    <column id="employer"
      link-entity="EmploymentRelationship.employmentID"
      entity="Employment">
      <title id="Employer.Title">Empleador</title>
    </column>
  </list>
</list>
</summary-page>
```

Figura 22. Página de resumen de asociación de entidades

4.6 Supresión de datos asociados

Cuando las entidades forman relaciones padre-hijo, si se suprime la entidad padre, también se suprimen todas las entidades hijo. Cuando se suprime una entidad que participa en una relación, de forma predeterminada se suprimen las relaciones de dicha entidad, pero las entidades relacionadas no.

Por ejemplo, suponga que los detalles de todas las personas de una unidad familiar se han recopilado y que se han creado entidades de persona y las relaciones entre las personas de la unidad familiar también se han capturado y las entidades de relación se han creado. Si el usuario elige eliminar una persona, las relaciones en las que participa dicha persona también se eliminarán, pero no se eliminará ninguna de las demás personas de la unidad familiar.

Este comportamiento predeterminado también se aplica al ejemplo de ingresos y empleo. Si el usuario elige eliminar un ingreso, cualquier relación de empleo para los ingresos se eliminará pero no se eliminará ninguna de las entidades de Empleo.

Es posible cambiar el comportamiento predeterminado al suprimir entidades asociadas de modo que también se eliminarán las entidades relacionadas con la entidad.

Para cambiar el comportamiento predeterminado, se puede añadir una anotación que contenga un elemento de documentación a la definición de una entidad de relación en el esquema de DS. Un elemento de documentación que contenga el texto "@curam.ieg.cascading.delete=true" indica que las entidades relacionadas deben suprimirse cuando se suprima la relación.

```
<xsd:element name="EmploymentRelationship">
  <xsd:annotation>
    <xsd:documentation>@curam.ieg.cascading.delete=true
  </xsd:documentation>
</xsd:annotation>
<xsd:complexType>
  <xsd:attribute name="employmentID" type="d:SVR_KEY" />
</xsd:complexType>
</xsd:element>
```

Figura 23. Esquema de supresiones en cascada

En el ejemplo de ingresos y empleo, si `curam.ieg.cascading.delete` se establece en `true` para el `EmploymentRelationship`, cuando se elimine entidad de ingresos también se eliminará cualquier entidad de Empleo asociada. La eliminación de las entidades de empleo utilizando este método no hace que se eliminen otras entidades de ingresos.

Capítulo 5. Formas eficientes de captura de datos

5.1 Introducción

En este capítulo se resaltan algunas de las características de IEG que permiten que la información se reúna de manera más eficaz y más intuitiva.

5.2 Preguntas de lista

En un ejemplo anterior, hemos visto un requisito para recopilar información de ingresos para las personas de una unidad familiar. A fin de recopilar sólo información de ingresos para las personas que realmente tienen ingresos, se ha añadido una pregunta a la página de Detalles de miembros de unidad familiar para indicar si la persona tiene ingresos o no.

IEG proporciona una alternativa a formular la misma pregunta booleana para varias entidades. Se puede utilizar una pregunta de lista para recopilar todas las respuestas al mismo tiempo.

Siguiendo con el ejemplo anterior donde se ha recopilado información acerca de las personas de la unidad familiar, se ha añadido el atributo `hasIncome` a la entidad de persona para indicar si se debe recopilar información de ingresos para la persona, como se indica a continuación:

```
<xs:element name="Person">
  <xs:complexType>
    ...
    <xs:attribute name="hasIncome" type="IEG_BOOLEAN"/>
  </xs:complexType>
</xs:element>
```

Figura 24. Esquema Has Income Person

Al igual que las preguntas, las preguntas de lista se deben añadir a un clúster. Las preguntas de lista difieren en que se debe especificar el tipo de las entidades que se visualizarán en la lista. El ID de la pregunta de lista corresponde al nombre del atributo booleano que debe establecerse si el usuario selecciona un elemento en la lista. Al igual que con las preguntas, una pregunta de lista debe tener una etiqueta para indicar la finalidad de la pregunta. Las preguntas de lista también deben tener un elemento de etiqueta de elemento. La etiqueta de elemento especifica qué atributo de las entidades se debe utilizar para identificar las entidades en la lista. En el ejemplo siguiente, se visualiza el nombre de los miembros de la unidad familiar para identificarlos.

```
<question-page id="AnyoneHaveIncome">
  ...
  <cluster>
    <list-question id="hasIncome" entity="Person">
      <label id="HasIncome.Label">
        <![CDATA[¿Qué personas tienen ingresos?]]>
      </label>
      <item-label>
        <label-element attribute-id="firstName"/>
      </item-label>
    </list-question>
  </cluster>
</question-page>
```

Figura 25. Pregunta de lista

Así, en lugar de añadir una pregunta en el bucle donde se recopilan los detalles de miembros de la unidad familiar, una vez que se han capturado los detalles de miembros de la unidad familiar, se puede visualizar una lista que contiene los miembros de la unidad familiar. A continuación, el usuario puede seleccionar los miembros que tienen ingresos.

Las preguntas de lista son especialmente útiles cuando se utilizan conjuntamente con un bucle for-each, que hace referencia a la pregunta que se ha establecido en la pregunta de lista de la expresión de criterios del bucle. Las preguntas de lista también se pueden utilizar con los tipos de entidad distintas de Person.

5.2.1 Selección única

Las preguntas de lista también se pueden utilizar cuando la selección debe ser mutuamente excluyente. Cuando el atributo single-select de una pregunta de lista está establecido en true, sólo se puede seleccionar uno de los elementos de la lista.

Si, por ejemplo, el requisito es indicar qué miembro de la unidad familiar es el cuidador principal, se puede añadir un atributo a la entidad de persona y se puede añadir al script una pregunta de selección única:

```
<xsd:element name="Person">
  <xsd:complexType>
    ...
    <xsd:attribute name="primaryCareGiver" type="IEG_BOOLEAN"/>
  </xsd:complexType>
</xsd:element>
```

Figura 26. Esquema de persona cuidador principal

```
<question-page id="PrimaryCareGiver" ...>
  ...
  <cluster>
    <list-question id="primaryCareGiver" entity="Person"
      single-select="true" criteria="age > 14">
      <label id="PrimaryCareGiver.Label">
        <![CDATA[¿Qué persona es el cuidador principal?]]>
      </label>
      <item-label>
        <label-element attribute-id="firstName" />
      </item-label>
    </list-question>
  </cluster>
</question-page>
```

Figura 27. Single-select List Question

La pregunta de lista anterior hará que se visualice una lista de los miembros de la unidad familiar que tienen más de 14 años con un botón de selección junto a cada persona, permitiendo así que sólo se seleccione una.

5.3 Preguntas de tabla de códigos

Si un atributo está definido en un esquema de DS como una tabla de códigos, cuando se visualiza la pregunta correspondiente, el comportamiento predeterminado es mostrar la pregunta como un menú desplegable. Sólo se puede seleccionar una respuesta en la lista desplegable.

Por ejemplo, si esto es un requisito para capturar el estado inicial de un miembro de la unidad familiar, se puede añadir una nueva definición de dominio para representar la tabla de código AddressState y se puede añadir un atributo para almacenar el estado inicial a la entidad de Persona de la siguiente manera:

```

...
<xsd:simpleType name="IEG_STATE_ADDRESS">
  <xsd:annotation>
    <xsd:appinfo>
      <D:options>
        <D:option name="code-table-name">AddressState</D:option>
      </D:options>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:restriction base="IEG_CODETABLE_CODE" />
</xsd:simpleType>
...

<xsd:element name="Person">
  ...
  <xsd:attribute name="homeState" type="IEG_STATE_ADDRESS" />

```

Figura 28. Tabla de códigos y atributo de estado

Se puede añadir al script una pregunta para capturar la información de estado inicial de la manera siguiente:

```

<question-page id="AboutYouPage" entity="Person">
...
  <cluster>
    <question id="homeState">
      <label id="State.Label">
        <![CDATA[Seleccione el estado inicial:]]>
      </label>
    </question>
  </cluster>

```

Figura 29. Pregunta de tabla de códigos de estado

Cuando el script se ejecuta, se visualiza la pregunta al usuario como una lista desplegable.

IEG también soporta la definición de preguntas de tabla de códigos de tal manera que el usuario pueda realizar selecciones múltiples.

Cuando una pregunta de tabla de códigos es de una sola selección, la respuesta a la pregunta se puede almacenar en un solo atributo de una entidad. Puesto que existen múltiples respuestas posibles en una pregunta de tabla de códigos de selecciones múltiples, se debe añadir una secuencia para almacenar todas las respuestas y se debe definir un nuevo tipo de entidad para representar las respuestas en la secuencia.

```

<xsd:element name="Person">
  <xsd:complexType>
    <xsd:sequence minOccurs="0">
      <xsd:element ref="State" minOccurs="0"
        maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

...
<xsd:element name="State">
  <xsd:complexType>
    <xsd:attribute name="stateCode" type="IEG_STATE_ADDRESS" />
  </xsd:complexType>
</xsd:element>

```

Figura 30. Entidad de estado

Una pregunta de tabla de códigos se convierte en una selección múltiple estableciendo el atributo multi-select de la pregunta en true. Cuando se añade una pregunta de tabla de códigos de selección múltiple, el clúster al que se está añadiendo la pregunta se debe correlacionar con el nuevo tipo de entidad que representa las respuestas a la pregunta. En nuestro ejemplo, el clúster se debe correlacionar con la entidad State. La página que contiene la pregunta de selección múltiple se debe correlacionar con la entidad que contiene la secuencia. En este ejemplo, la página se debe correlacionar con la entidad de persona. Por último, para que sean visibles varias opciones en una pregunta de tabla de códigos de selección múltiple, se debe añadir un diseño a la pregunta. El diseño debe especificar el número de filas visibles para la pregunta. Si el número de opciones disponibles para la pregunta excede el número de filas especificadas en el diseño, se añadirá una barra de desplazamiento a la pregunta.

```

<question-page id="AboutYouPage" entity="Person">
...
  <cluster entity="State">
    <question id="stateCode" multi-select="true">
      <label id="State.Label">
        <![CDATA[Selecione los estados en que ha vivido:]]>
      </label>
      <layout>
        <num-rows>4</num-rows>
      </layout>
    </question>
  </cluster>

```

Figura 31. Pregunta de tabla de códigos de selección múltiple

Cuando se ejecuta el script, la pregunta se visualiza al usuario como una lista de descripciones de tabla de códigos con un recuadro de selección para cada elemento.

5.4 Elementos condicionales

Los scripts IEG pueden tener varios elementos condicionales diferentes: secciones, páginas o clústeres. Los elementos condicionales se pueden mostrar u ocultar basándose en las respuestas de las páginas anteriores o en datos llenados previamente en el DS.

5.4.1 Secciones condicionales

Es posible eliminar secciones de una ejecución de script evaluando una expresión a iniciarse la ejecución: si la sección no es visible, no se listará en el panel de secciones y la expresión no se podrá volver a evaluar durante la ejecución de script.

Mediante el uso de un DS llenado previamente como se describe en 8.3, "Llenado previo de scripts con datos capturados", en la página 55, podemos establecer un distintivo en una entidad dependiendo de las circunstancias externas al script. Imaginemos que tenemos una entidad denominada IntakeInformation que tiene un atributo booleano "collectIncomeInformation". Podemos especificar una sección Income en nuestro script:

```
...
<section visible="IntakeInformation.collectIncomeInformation==true">
  ...
</section>
...
```

Figura 32. Atributo visible de una sección

Esto ocultará la sección Income si el atributo "collectIncomeInformation" es false, como si la sección no estuviera presente en la definición de script.

Si una sección debe estar habilitada o inhabilitada en función de las respuestas de las secciones anteriores, es posible acomodar todas las páginas de una sección en una sola condición. A diferencia del atributo visible, esta condición se evaluará siempre que se encuentre la sección, lo que significa que es posible retroceder y cambiar una respuesta que afecta a la navegación de una sección. La sección seguirá apareciendo en el panel de secciones, pero aparecerá sombreada para que el usuario no puede pulsar en ella.

Se puede modificar el ejemplo anterior para que la pregunta "collectIncomeInformation" se formule al principio del script. Entonces la sección Income se puede modificar de la manera siguiente:

```
...
<section>
  <condition
    expression="IntakeInformation.collectIncomeInformation">
    ...
  </condition>
</section>
...
```

Figura 33. Sección condicional

5.4.2 Páginas condicionales

Las páginas se pueden visualizar o no en función del valor de una expresión de condición. Los bucles también se pueden reiniciar en estas condiciones.

La sección condicional mencionada anteriormente, donde una condición reinicia el contenido de toda la sección, es un ejemplo de páginas condicionales.

5.4.3 Clústeres condicionales

Los clústeres también se pueden reiniciar en un elemento de condición. Si la expresión del elemento de condición no hace referencia a ninguna de las preguntas de la misma página, el clúster es un clúster condicional estático. Esto se debe a que se puede determinar antes de que se visualice la página si se debe mostrar el clúster o no.

Por ejemplo, si se ha reunido información sobre los miembros de la unidad familiar, puede que desee añadir otra página para solicitar más detalles personales, incluyendo si la persona está embarazada. Se deberá añadir un atributo nuevo isPregnant a la entidad de persona para almacenar esta información:

```

<xsd:element name="Person">
  <xsd:complexType>
    ...
    <xsd:attribute name="isPregnant" type="IEG_BOOLEAN"/>

```

Figura 34. Atributo de persona adicional

Por supuesto, esta pregunta es sólo aplicable si el sexo es femenino. Por lo tanto, el clúster puede reiniciarse en una condición y sólo se mostrará si la expresión de condición se evalúa en true. La nueva página de detalles de la persona adicionales se puede definir de la manera siguiente:

```

<question-page id="AboutTheClientContinued" entity="Person" ...>
  <condition expression="Person.gender=='SX2'">
    <cluster>
      <question id="isPregnant" mandatory="true">
        <label id="IsPregnant.Label">
          ¿Está embarazada?
        </label>
        <help-text id="IsPregnant.HelpText">
          ¿Está embarazada?
        </help-text>
      </question>
    </cluster>
  </condition>
</question-page>

```

Figura 35. Clúster condicional estático

De forma alternativa, si alguna de las preguntas a las que se hace referencia en la expresión de condición se encuentra en la misma página, entonces el clúster es un clúster dinámicamente condicional. Esto significa que el clúster se visualizará y se ocultará a medida que el usuario cambie las respuestas a las preguntas de la página. Esta característica dinámica de IEG requiere que esté habilitado JavaScript en el navegador. Las expresiones del clúster dinámicamente condicional no pueden hacer referencia a funciones personalizadas, ya que las expresiones se evalúan sin realizar una llamada al servidor.

Sin cambiar el esquema de DS, si el ejemplo anterior se cambia de forma que el clúster condicional se defina en la misma página que la pregunta de sexo, el clúster será un clúster dinámicamente condicional.

```

<question-page id="AboutTheClient" entity="Person" ...>
  ...
  <cluster>
    <title id="DetailsCluster.Title">
      <![CDATA[Detalles personales]]>
    </title>
    ...
    <question id="gender" mandatory="true">
      <label id="Gender.Label">
        <![CDATA[Sexo:]]>
      </label>
    </question>
    ...
    <condition expression="Person.gender=='SX2'">
      <cluster>
        <question id="isPregnant" mandatory="true">
          <label id="IsPregnant.Label">
            <![CDATA[¿Está embarazada?]]>
          </label>
        </question>
      </cluster>
    </condition>
  </question-page>

```

Figura 36. Clúster dinámicamente condicional

La pregunta sobre el embarazo aparecerá o desaparecerá dinámicamente cuando cambie el valor seleccionado para el sexo. El comportamiento dinámico en una página pueden desencadenarlo los campos de texto, los campos de fecha, los recuadros de selección, los botones de selección y los elementos de selección. El comportamiento dinámico no lo puede desencadenar la respuesta a una pregunta de varias selecciones o una matriz de pregunta, debido a las restricciones de la sintaxis de expresión.

Se debe tener en cuenta que sólo se permite un nivel de condición en torno a un clúster, es decir los clústeres condicionales no se pueden anidar en otras condiciones. La expresión de condición para un clúster dinámicamente condicional puede hacer referencia a preguntas en la misma página que a su vez están definidas en un cluster dinámicamente condicional. Esto crea una dependencia en cascada entre los clústeres.

5.5 Matrices de preguntas

Las preguntas de lista presentadas en 5.2, "Preguntas de lista", en la página 27 hacen la misma pregunta booleana acerca de un grupo de entidades. Es posible plantear la misma pregunta de tabla de códigos para un grupo de entidades utilizando matrices de pregunta.

Una matriz de pregunta mostrará una lista de preguntas basándose en una tabla de códigos y para cada uno de estos valores de la tabla de códigos y cada entidad, se visualizará un recuadro de selección para permitir al usuario seleccionar todos los valores que se aplican a una entidad determinada.

Por ejemplo, suponga que hay requisito de capturar los posibles niveles de abuso de sustancias para cada miembro de la unidad familiar, se puede añadir una nueva definición de dominio para representar la tabla de códigos SubstanceAbuse y se puede añadir un atributo para almacenar el nivel de abuso de sustancias a la entidad Person, como se indica a continuación:

```
<xsd:simpleType name="IEG_SUBSTANCEABUSE">
  <xsd:annotation>
    <xsd:appinfo>
      <D:options>
        <D:option name="code-table-name">SubstanceAbuse</D:option>
      </D:options>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:restriction base="IEG_CODETABLE_CODE" />
</xsd:simpleType>

<xsd:element name="Person">
  <xsd:complexType>
    ...
    <xsd:attribute name="substanceAbuse"
      type="IEG_SUBSTANCEABUSE" />
  </xsd:complexType>
</xsd:element>
```

Figura 37. Atributo de abuso de sustancias

Entonces la matriz de pregunta se define como una pregunta de lista normal y sólo el hecho de que se basa en una tabla de códigos en lugar de un valor booleano hará que se visualice de forma diferente.

```

...
<list-question entity="Person" id="substanceAbuse"
  criteria="age > 14">
  <label id="SubstanceAbuse.Label">
    <![CDATA[Abuso de sustancias:]]>
  </label>
  <item-label>
    <label-element attribute-id="firstName" />
  </item-label>
</list-question>

```

Figura 38. Ejemplo de código de matriz de pregunta

El ejemplo anterior de una matriz de pregunta, que recopila información de abuso de sustancias sobre varios miembros de la unidad familiar, se mostrará en el reproductor IEG como una matriz en la que cada fila corresponde a una descripción de tabla de códigos y cada columna a una persona.

5.6 Navegación de Fast Path

De forma predeterminada, cuando un usuario vuelve a iterar por un script, todas las páginas se vuelven a visualizar, lo que puede ser una tarea ardua, especialmente en unidades familiares grandes. La navegación de Fast Path permite a los usuarios finales pasar más rápidamente por los scripts IEG saltando automáticamente las páginas de bucle o condicionales que ya se han respondido.

Esta funcionalidad es opcional y está desactivada de forma predeterminada. Se puede activar en bucles y condiciones (para activar la navegación Fast Path, consulte la guía *Authoring Scripts using Intelligent Evidence Gathering (IEG)*).

La primera vez que se encuentra un elemento de vía de acceso rápida, éste se comporta de forma normal. Cuando el usuario navega por el script posteriormente, sólo se visualizarán las nuevas páginas dentro de estos elementos de vía de acceso rápida. Las páginas que se han mostrado anteriormente se saltarán ahora. Esta funcionalidad no impide editar los datos a través de los enlaces de edición de una página de resumen si es necesario.

Fast Path puede utilizarse en los escenarios siguientes:

- Pregunta de lista que controla un bucle
- Criterios de elegibilidad
- Condiciones de Fast Path
- Condición en bucle Fast Path

5.6.1 Pregunta de lista que controla un bucle

Utilizando la misma pregunta de lista que se describe en 5.2, “Preguntas de lista”, en la página 27, queremos recopilar información de ingresos para las personas de una unidad familiar. Vamos a utilizar un bucle de vía de acceso rápida anidado como se describe en el ejemplo siguiente:

```

...
<loop loop-type="for-each" entity="Person"
  criteria="hasIncome==true" fast-path="true">
  <loop loop-type="while" loop-expression="hasMoreIncome"
    entity="Income">
    <question-page id="IncomePage" entity="Income"
      show-person-tabs="true">
      <title id="IncomePage.Title">
        <![CDATA[Detalles de ingresos]]>
      </title>
      <cluster>
        <title id="IncomeDetails.Title">
          <![CDATA[Detalles de ingresos]]>
        </title>
        <question id="type">
          <label id="Type.Label">
            <![CDATA[Tipo:]]>
          </label>
        </question>
        <question id="amount">
          <label id="Amount.Label">
            <![CDATA[Importe:]]>
          </label>
        </question>
        <question id="hasMoreIncome"
          control-question="true"
          control-question-type="IEG_BOOLEAN">
          <label id="ContinueQuestion.Label">
            <![CDATA[Does %1s have any more income?]]>
            <argument id="Person.firstName" />
          </label>
        </question>
      </cluster>
    </question-page>
  </loop>
</loop>

```

Figura 39. Ejemplo de código de pregunta de lista de Fast Path que controla un bucle

La primera vez que se encuentra la pregunta de lista, las páginas que siguen al bucle recopilarán los ingresos para las personas que se han seleccionado. A continuación, cuando se vuelva a visitar la página que contiene la pregunta de lista, puede ocurrir lo siguiente:

- Si los recuadros de selección no se han modificado, al pulsar Siguiente se saltará el bucle de ingresos y se visualizará la página posterior al bucle.
- Si algunos de los recuadros están deseleccionados, al pulsar Siguiente se suprimirán los ingresos correspondientes a las personas que no se han seleccionado, se saltará el bucle de ingresos y se visualizará la página posterior al bucle.
- Si se seleccionan recuadros de selección nuevos, al pulsar Siguiente se saltarán las páginas de ingresos existentes, se mostrarán páginas de ingresos nuevas para las personas recién seleccionadas y, a continuación, se visualizará la página posterior al bucle.
- Si se seleccionan recuadros de selección nuevos y otros están deseleccionados, al pulsar Siguiente se suprimirán los ingresos correspondientes a las personas que estaban deseleccionadas, se saltarán las páginas de ingresos existentes, se mostrarán páginas de ingresos nuevas para las personas recién seleccionadas y, a continuación, se visualizará la página posterior al bucle.

5.6.2 Criterios de elegibilidad

Basándonos en el escenario anterior, se pueden filtrar las personas que se visualizarán en la pregunta de lista (el bucle no necesita modificarse). Sólo las personas de más de 18 será aptas para entrar los ingresos, de modo que se añade un criterio a la pregunta de lista. Al reiterar por el script, es posible que haya personas que ya no coinciden con los criterios y, por lo tanto, no aparecen en la lista.

```

...
<list-question id="hasIncome" entity="Person" criteria="age > 18">
  <label id="HasIncome.Label">
    <![CDATA[¿Qué personas tienen ingresos?]]>
  </label>
  <item-label>
    <label-element attribute-id="firstName" />
  </item-label>
</list-question>

```

Figura 40. Código de ejemplo de pregunta de lista de Fast Path con criterios de elegibilidad que controlan un bucle

De esta forma se comportará como se menciona en el escenario anterior, pero si se modifica la fecha de nacimiento de una persona, se producirá lo siguiente:

- Si la persona se vuelve inelegible (menos de 18) y se habían entrado ingresos, los ingresos correspondientes se suprimirán automáticamente tan pronto como se envíe la nueva fecha de nacimiento.
- Si la persona se vuelve elegible (más de 18), se visualizará en la pregunta de lista (pero no se seleccionará) la próxima vez que se visualice la página de preguntas de lista.

5.6.3 Condiciones de Fast Path

Podemos solicitar detalles de embarazo a los miembros femeninos de la unidad familiar utilizando una página condicional. Si la condición se define como vía de acceso rápida, los detalles de embarazo se ocultarán cuando se vuelva a iterar por los miembros de la unidad familiar ya que las páginas de la condición sólo se visualizarán al volver a iterar por el script si la condición se ha evaluado anteriormente como falsa y algo ha cambiado de modo que ahora la condición se evalúa como verdadera.

```

...
<question-page id="AboutYouPage" entity="Person">
  <title id="PrimaryPersonPage.Title">
    <![CDATA[Acerca de usted]]>
  </title>
  <cluster>
    <title id="DetailsCluster.Title">
      <![CDATA[Detalles personales]]>
    </title>
    <question id="firstName" mandatory="true">
      <label id="FirstName.Label">
        <![CDATA[Nombre:]]>
      </label>
    </question>
    <question id="middleName">
      <label id="MiddleName.Label">
        <![CDATA[Segundo nombre:]]>
      </label>
    </question>
    <question id="lastName">
      <label id="lastName.Label">
        <![CDATA[Apellidos:]]>
      </label>
    </question>
    <question id="gender" mandatory="true">
      <label id="Gender.Label">
        <![CDATA[Sexo:]]>
      </label>
    </question>
    <question id="dateOfBirth" mandatory="true">
      <label id="DateOfBirth.Label">
        <![CDATA[Fecha de nacimiento:]]>
      </label>
    </question>
  </cluster>
</question-page>
<condition expression="Person.gender=='SX2'"
  fast-path="true">
  <question-page id="PregnancyPage" entity="Person">
    <title id="PregnancyPage.Title">
      <![CDATA[Acerca de usted: embarazo]]>
    </title>
    <cluster>
      <title id="DetailsCluster.Title">
        <![CDATA[Detalles personales acerca de su embarazo]]>
      </title>
      <question id="isPregnant" >
        <label id="IsPregnant.Label">
          <![CDATA[¿Está embarazada?]]>
        </label>
      </question>
    </cluster>
  </question-page>
</condition>

```

Figura 41. Ejemplo de código de condiciones de Fast Path

Al editar los detalles personales, puede ocurrir lo siguiente:

- Si no se ha realizado ningún cambio en el género, al pulsar en Siguiente se saltará la condición, tanto si se ha visualizado la primera vez como si no.
- Si el género ha cambiado de Hombre a Mujer, al pulsar en Siguiente se mostrará la página condicional para especificar detalles de embarazo.
- Si el género ha cambiado de Mujer a Hombre, al pulsar Siguiente se suprimirán los detalles de embarazo y se visualizará la página después de la condición.

5.6.4 Condición en bucle Fast Path

Cuando se define una condición dentro de un bucle Fast Path, éste se comportará igual que cuando se utiliza un criterio en el bucle en lugar de anidar una condición, con la excepción siguiente: si la condición es verdadera, la página contenida en la condición no se puede visualizar porque el bucle no tiene una nueva iteración que mostrar y, por lo tanto, se omite. Si la condición es falsa, la página y los datos asociados no se suprimirán porque la condición no se vuelve a evaluar. Por lo tanto, se recomienda utilizar un criterio en el bucle en lugar de una condición.

```
...
<loop loop-type="for-each" entity="Person"
  fast-path="true">
  <condition expression="Person.hasIncome==true">
    <loop loop-type="while" loop-expression="hasMoreIncome"
      entity="Income">
      <question-page id="IncomePage" entity="Income"
        show-person-tabs="true">
        <title id="IncomePage.Title">
          <![CDATA[Detalles de ingresos]]>
        </title>
        <cluster>
          <title id="IncomeDetails.Title">
            <![CDATA[Detalles de ingresos]]>
          </title>
          <question id="type">
            <label id="Type.Label">
              <![CDATA[Tipo:]]>
            </label>
          </question>
          <question id="amount">
            <label id="Amount.Label">
              <![CDATA[Importe:]]>
            </label>
          </question>
          <question id="hasMoreIncome"
            control-question="true"
            control-question-type="IEG_BOOLEAN">
            <label id="ContinueQuestion.Label">
              <![CDATA[¿Tiene %1s más ingresos?]]>
              <argument id="Person.firstName" />
            </label>
          </question>
        </cluster>
      </question-page>
    </loop>
  </condition>
</loop>
```

Figura 42. Ejemplo de condición en código de bucle Fast Path

5.7 Supresión implícita

Siempre que es posible, el motor IEG intenta suprimir datos tan pronto como averigua que ya no son importantes.

Si el usuario modifica explícitamente una respuesta (a través de una pregunta regular, una lista de preguntas o un establecimiento de atributo, pero no a través de una llamada a función personalizada), el motor detecta si esta respuesta se utiliza en una expresión de condición, un criterio de lista de preguntas o un criterio de bucle. Si ese es el caso, la expresión o los criterios se vuelven a evaluar y, si se han convertido en falsos, se eliminan las páginas correspondientes y los datos asociados se suprimen sin tener que pasar por el script para encontrar las expresiones o los criterios.

Capítulo 6. Otras consideraciones sobre el desarrollo de script

6.1 Introducción

Las diferentes construcciones que se han presentado hasta ahora satisfacen muchas necesidades de recopilación de pruebas diferentes, pero puede haber situaciones que requieran funciones adicionales, como la posibilidad de visualizar datos en una modalidad de sólo lectura o de invocar funciones externas. En este capítulo se detallan estos elementos.

Este capítulo también incluye algunos temas que deben tenerse en cuenta al mantener scripts IEG, poner scripts bajo el control de origen y cargar scripts en la base de datos.

6.2 Visualización de datos como de sólo lectura

A veces es necesario mostrar al usuario las respuestas a algunas preguntas forma que no se puedan modificar. Esto ya sucede en el caso de las páginas de resumen donde los usuarios pueden revisar las respuestas y utilizar el botón de retroceso o editar enlaces para modificarlas.

En una página de preguntas, se puede establecer un atributo booleano "sólo lectura" en true, lo que indica que todas las preguntas que se muestran en la página no se pueden editar.

Existe un mecanismo más sofisticado: se pueden utilizar atributos de "expresión de sólo lectura" en elementos de script diferentes (secciones, todos los tipos de páginas, clústeres, preguntas y preguntas de lista). Si la expresión se evalúa como verdadera (true), esto se aplicará a todas las preguntas contenidas en el elemento. En su forma más simple, la expresión será "true" si el elemento necesita ser de sólo lectura de forma incondicional. En una página de resumen, el resultado es que los enlaces para añadir, editar y suprimir no se visualizan.

En el caso de expresión de sólo lectura definida para el clúster, los elementos de script de pregunta de lista y pregunta, si alguna de las preguntas a las que se hace referencia en la expresión está en la misma página que el elemento de script, se habilitan o inhabilitan dinámicamente, en lugar de ser simplemente de sólo lectura. Esto significa que las preguntas se habilitarán e inhabilitarán a medida que el usuario cambie las respuestas a otras preguntas de la página. Cuando la expresión de sólo lectura de un clúster haga referencia a una pregunta de la misma página, todas las preguntas contenidas en el clúster se habilitarán e inhabilitarán. Esta característica dinámica de IEG requiere que esté habilitado JavaScript en el navegador. Las expresiones para habilitar e inhabilitar dinámicamente las preguntas pueden no hacer referencia a funciones personalizadas, ya que las expresiones se evalúan sin realizar una llamada al servidor.

Las expresiones de sólo lectura dinámicas también pueden hacer referencia a las preguntas de la misma página que a su vez se habilitan e inhabilitan dinámicamente. Esto crea una dependencia en cascada entre las preguntas. Se deberá tener cuidado al definir expresiones con dependencias en cascada porque IEG no tiene en cuenta si las preguntas a las que se hace referencia en la expresión de sólo lectura está habilitadas o no, sólo el valor de la pregunta. Esto puede ser confuso para el usuario porque puede que no sea aparente qué está controlando la habilitación e inhabilitación de una pregunta.

Cuando se visualiza una pregunta, si el atributo de almacén de datos correspondiente tiene un valor, éste se mostrará incluso si la pregunta está inicialmente inhabilitada. Entonces el usuario puede habilitar la pregunta y puede cambiar la respuesta. Si la pregunta está inhabilitada, su valor se volverá a establecer en el valor que tenía cuando se visualizó inicialmente. Cuando se envía una página, el atributo de almacén no se actualizará a menos que la pregunta esté habilitada. Por lo tanto, si la página se vuelve a visualizar, el valor original del atributo de almacén de datos se mostrará de nuevo.

No es posible marcar una pregunta como obligatoria si también tiene una expresión de sólo lectura dinámica expresión en la propia pregunta o en uno de sus elementos padre.

La habilitación e inhabilitación dinámicas de elementos de script no se soporta en páginas de relaciones.

La información recopilada en bucles se puede visualizar en las páginas de resumen utilizando listas, pero también es posible utilizar esta construcción de lista en páginas normales sin necesidad de especificar una expresión de sólo lectura en uno de los elementos que reinician la lista. La única diferencia con las listas de resumen es que los enlaces no están permitidos.

Otra posibilidad es hacer que todo un script sea de sólo lectura. Esto es útil, por ejemplo, si un asistente social necesita revisar un script sin poder cambiar ninguna de las respuestas. El script se establece en sólo lectura mediante IEGRuntimeAPI estableciendo un distintivo de sólo lectura en la ejecución de script, tal como se muestra a continuación:

```
...
//Establecer distintivo de sólo lectura.
IEGRuntime runtimeAPI = new IEGRuntime();
IEGScriptExecutionID runtimeExecID = new IEGScriptExecutionID();
runtimeExecID.executionID = execution.getExecutionID();
IEGReadOnlyFlag readOnlyFlag = new IEGReadOnlyFlag();
readOnlyFlag.readOnlyFlag = true;
runtimeAPI.setReadOnlyFlag(runtimeExecID, readOnlyFlag);
...
```

Figura 43. Establecimiento del distintivo de sólo lectura en una ejecución de script

6.3 Invocación de funciones externas utilizando expresiones

Se pueden encontrar expresiones en varios lugares de un script para definir el comportamiento de los bucles, las condiciones, etc. Vea el apéndice de sintaxis de expresiones en la guía Authoring Scripts using Intelligent Evidence Gathering (IEG) para realizar consultas.

Estas expresiones pueden hacer referencia a respuestas, pueden combinarlas utilizando diversos operadores e incluso pueden llamar a las funciones (excepto cuando se utilizan en clústeres condicionales dinámicos porque estas expresiones se evalúan en el navegador).

Las funciones descritas anteriormente se denominan de Funciones personalizadas y se definen utilizando código Java™. En función de la utilización, pueden ser de dos tipos:

- Funciones personalizadas que pueden tomar parámetros (posiblemente realizando una llamada a una función externa) y devolverán un valor. No modificarán el contenido del DS. Se utilizan en la mayoría de expresiones.
- Cuando el objetivo es actualizar el contenido del DS, se puede utilizar la función personalizada en un elemento autónomo: comentario-emergente. El valor devuelto es irrelevante (pero debe ser un valor booleano). La función personalizada no debe actualizar valores que se han contestado antes del comentario emergente. Esto se debe a que el motor IEG no está consciente de las actualizaciones realizadas fuera del contexto del script y, por lo tanto, no puede tomar las acciones necesarias para las actualizaciones.

Ejemplos reales que pueden necesitar la invocación de funciones externas son la validación de un código postal de EE.UU. que un usuario ha proporcionado y la población de un campo de estado basado en un código postal proporcionado. Ahora vamos a demostrar esos dos usos diferentes.

El esquema de DS se tendrá que ampliar para añadir los 2 atributos siguientes a la entidad de persona, como se indica a continuación:

```
<xsd:attribute name="state" type="IEG_STRING"/>
<xsd:attribute name="zipCode" type="IEG_STRING"/>
```

Figura 44. Atributos de persona adicionales en el esquema de DS

Primero intentemos validar un código postal en un estado (es una implementación simplista): un código postal debe tener una longitud de cinco dígitos y los tres primeros dígitos indicarán el estado.

La página de detalles personales mencionada anteriormente y la página de resumen correspondiente se pueden modificar con 2 preguntas obligatorias adicionales: estado y código postal:

```
<question id="state" mandatory="true">
  <label id="State.Label">
    Estado:
  </label>
  <help-text id="State.HelpText">
    El estado en que vive
  </help-text>
</question>
<question id="zipCode" mandatory="true">
  <label id="ZipCode.Label">
    Código postal:
  </label>
  <help-text id="ZipCode.HelpText">
    Su código postal
  </help-text>
</question>
```

Figura 45. Preguntas de estado y código postal en la definición de script

Entonces la función personalizada que realizar la validación se debe crear como una clase Java en el paquete `curam.rules.functions`:

```

...
public class CustomFunctionValidateZipCode extends CustomFuncor {

    public Adaptor getAdaptorValue(final RulesParameters rp)
    throws AppException, InformationalException {

        final List<Adaptor> parameters = getParameters();
        final String zipCode =
            ((StringAdaptor) parameters.get(0)).getStringValue(rp);
        final String state =
            ((StringAdaptor) parameters.get(1)).getStringValue(rp);
        boolean valid = false;

        if (zipCode.length() == 5) {
            final String prefix = zipCode.substring(0, 3);
            //buscar los prefijos de estado
            if (prefix.equals("100")
                && state.equalsIgnoreCase("Nueva York")) {
                valid = true;
            }
            if (prefix.equals("900")
                && state.equalsIgnoreCase("California")) {
                valid = true;
            }
        }

        return AdaptorFactory.getBooleanAdaptor(Boolean.valueOf(valid));
    }
}

```

Figura 46. Función personalizada para validar el código postal

Los metadatos siguientes para la función personalizada se deben insertar en <sucomponents>/rulesets/functions/CustomFunctionMetaData.xml:

```

<CustomFuncor name="CustomFunctionValidateZipCode">
  <parameters>
    <parameter>
      curam.util.rules.funcor.Adaptor$StringAdaptor
    </parameter>
    <parameter>
      curam.util.rules.funcor.Adaptor$StringAdaptor
    </parameter>
  </parameters>
  <returns>curam.util.rules.funcor.Adaptor$BooleanAdaptor</returns>
</CustomFuncor>

```

Figura 47. Metadatos de función personalizada

Consulte la Guía de codificación de reglas de Cúram para obtener más detalles sobre la definición de las funciones personalizadas.

En nuestro ejemplo, la función personalizada ValidateZipCode no accede a una base de datos externa para buscar el estado correspondiente. Idealmente, debe realizar esa búsqueda y, a continuación, comprobar el estado devuelto con el estado que se ha entrado. Para simplificarlo, más arriba sólo se han codificado dos prefijos de código postal.

Entonces la validación se insertará en la página de detalles personales:

```
<validation
  expression="ValidateZipCode(Person.zipCode, Person.state)">
  <message id="InvalidZipCode">
    El código postal no es válido.
  </message>
</validation>
```

Figura 48. Validación de código postal en la definición de script

Cuando el usuario pulsa Siguiente, las respuestas a las preguntas de código postal y estado se pasan a la función personalizada, que devolverá "true" si las respuestas son válidas. Entonces se visualizará la página siguiente.

Si la función personalizada devuelve "false", se visualiza el mensaje especificado en la validación en la parte superior de la página de detalles de la persona, bloqueando el acceso a la página siguiente hasta que se someten respuestas válidas.

La función personalizada no tiene ningún efecto secundario porque no modifica nada. Sólo realiza una operación basada en los parámetros y devuelve un resultado.

También sería posible eliminar el distintivo obligatorio en las dos nuevas preguntas y validar las respuestas sólo si han sido proporcionado ambas. Entonces será necesario cambiar la expresión de validación por lo siguiente utilizando la función personalizada lista para utilizarse `isNotNull` que comprueba si el parámetro proporcionado es nulo:

```
"not(isNotNull(Person.zipCode) and isNotNull(Person.state))
  or ValidateZipCode(Person.zipCode, Person.state)"
```

Figura 49. Expresión de validación alternativa

De forma alternativa, es posible llenar la pregunta de estado dado el código postal. Para ello, la página de detalles de la persona sólo solicitará el código postal (con el distintivo obligatorio) y la página de resumen mostrará el estado y el código postal.

Se debe definir la siguiente función personalizada:

```

...
public class CustomFunctionpopulateState extends CustomFuncutor {

    public Adaptor getAdaptorValue(final RulesParameters rp)
    throws AppException, InformationalException {

        final IEG2Context ieg2Context = (IEG2Context) rp;
        final long rootEntityID = ieg2Context.getRootEntityID();
        String schemaName;
        //schemaName tiene que codificarse o recuperarse fuera de IEG
        Datastore ds = null;
        try {
            ds =
                DatastoreFactory.newInstance().openDatastore(
                    schemaName);
        } catch (NoSuchSchemaException e) {
            throw new AppException(IEG.ID_SCHEMA_NOT_FOUND);
        }

        Entity applicationEntity = ds.readEntity(rootEntityID);

        Entity personEntity =
            applicationEntity.getChildEntities(
                ds.getEntityType("Person"))[0];
        String zipCode = personEntity.getAttribute("zipCode");
        String state = "Unknown";
        final String prefix = zipCode.substring(0, 3);
        //buscar los prefijos de estado
        if (prefix.equals("100")) {
            state = "Nueva York";
        }
        if (prefix.equals("900")) {
            state = "California";
        }
        personEntity.setAttribute("state", state);
        personEntity.update();
        return AdaptorFactory.getBooleanAdaptor(new Boolean(true));
    }
}

```

Figura 50. Función personalizada para llenar el estado

Y los metadatos:

```

<CustomFuncutor name="CustomFunctionpopulateState">
    <returns>curam.util.rules.funcutor.Adaptor$BooleanAdaptor</returns>
</CustomFuncutor>

```

Figura 51. Metadatos de función personalizada

Entre la página de detalles de la persona y la página de resumen, se debe insertar un elemento de llamada para llamar a esta función personalizada, como se indica a continuación:

```

<callout id="populateAddress" expression="populateState()"/>

```

Figura 52. Llamada para llenar el estado en la definición de script

Esta vez, la función personalizada modificará el DS llenando el estado en la entidad de persona. El contexto contiene el ID de entidad raíz y el ID de ejecución, lo que facilita la actualización del DS. Si la llamada se encuentra en un bucle, el contexto también contiene el ID de entidad actual.

6.4 Reutilización de scripts

Es posible desglosar una definición de script en varios archivos proporcionando así un mecanismo de reutilización.

Para lograr esto, una definición de script tendrá que hacer referencia a los subscripts. Cada uno de estos subscripts será un script autónomo que se puede ejecutar de forma independiente.

A continuación se muestra un ejemplo de un script que puede utilizarse como un subscript:

```
<?xml version="1.0" encoding="UTF-8"?>
<ieg-script ...>
  <identifier id="Subscript" scriptversionnumber="V1" type="Test" />
  <question-page ...>
    ...
  </question-page>
  ...
</ieg-script>
```

Figura 53. Subscript que contiene páginas

El script del fragmento de código de ejemplo anterior puede incluirse en otro script como subscript, como se indica a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>
<ieg-script ...>
  <identifier id="Script" scriptversionnumber="V1" type="Test" />
  <section>
    <ieg-sub-script>
      <identifier id="Subscript"
        scriptversionnumber="V1" type="Test" />
    </ieg-sub-script>
  </section>
  <section>
    ...
  </section>
  ...
</ieg-script>
```

Figura 54. Inclusión de un subscript en un script

El posible punto de inserción de un subscript en un script puede ser el siguiente:

- Si el script contiene secciones y el subscript también contiene secciones, el subscript se tendrá que insertar en el nivel superior, bajo el elemento `ieg-script` padre.
- Si el script contiene secciones y el subscript contiene secciones, el subscript se tendrá que insertar en una sección del script padre.
- Si el script no contiene secciones, el subscript no puede contener secciones. Se insertará en el nivel superior, bajo el elemento `ieg-script`.

Otra limitación a tener en cuenta es que un subscript sólo puede aparecer una vez en un script porque los ID de página deben ser exclusivos en el script resultante.

Tenga en cuenta que un script puede utilizarse como un subscript en otros lugares. Al modificar scripts, asegúrese de que los scripts de referencia se vuelven a probar para asegurarse de que los cambios no tienen un impacto no deseado.

6.5 Control de origen y versiones

Las definiciones de script IEG se almacenan en la base de datos. Al editar un script IEG utilizando el Editor de IEG, el script se edita en su lugar y se actualiza directamente en la base de datos. Las definiciones de script IEG son artefactos de desarrollo y desde un punto de vista de gestión de configuración de software es importante que estos artefactos se coloquen bajo el control de origen como se haría con otros artefactos.

Es posible descargar una definición de script de las pantallas de administración de script IEG. Cuando se elige la opción de descargar un script, primero se recupera el script de la base de datos, a continuación se recuperan del almacén de recursos los archivos de propiedades asociados con la definición de script y las propiedades textuales se "inyectan" en la definición de script antes de que éste quede disponible. Sin embargo descargar un script de este modo no proporciona todos los recursos que pueden estar asociados con una definición de script. Por ejemplo, no proporciona archivos de propiedades en varias configuraciones regionales y no proporciona imágenes e iconos. Consulte el apéndice *Compliance* de la guía del desarrollador de *Authoring Scripts using Intelligent Evidence Gathering (IEG)* para obtener más información sobre la representación de base de datos de un script IEG.

Al llenar la base de datos con definiciones de script, es importante tener en cuenta las diferencias en la funcionalidad entre la importación de un script a través de las pantallas de administración de script IEG y la carga de definiciones de script a través de archivos DMX.

Capítulo 7. Integración de IEG en una aplicación Cúram

7.1 Introducción

Este capítulo describe cómo IEG puede integrarse en una aplicación. IEG se puede integrar de dos maneras: abriendo el reproductor en una pestaña o en un diálogo modal. Las tareas de integración que se tratan aquí incluyen la creación de la ejecución de script, el establecimiento de las páginas de finalización y salida, la ejecución en una pestaña, la ejecución en un modal, la limpieza de datos de aplicación y la reanudación de scripts.

7.2 Creación de una ejecución de script

Se recomienda que, antes de abrir el reproductor IEG desde una aplicación, se cree la ejecución de script utilizando la API pública. El ID de ejecución se puede pasar entonces al reproductor.

El fragmento de código de ejemplo siguiente muestra la creación de una ejecución de script utilizando la API pública:

```
...  
  
// crear la ejecución de script  
final IEGRuntime runtimeAPI = new IEGRuntime();  
final IEGScriptExecutionIdentifier executionIdentifier =  
    runtimeAPI.createScriptExecution(iegScriptID, schemaName);
```

Figura 55. Creación de una ejecución de script

7.3 Especificación de un URL de redirección

Los atributos `finish-page` y `quit-page` en un script IEG indican a qué URL se debe redirigir cuando se sale del reproductor IEG. De este modo proporcionan una conexión entre el reproductor IEG y una aplicación. Estos atributos se detallan en el capítulo IEG Script Element Reference de la guía del desarrollador *Authoring Scripts using Intelligent Evidence Gathering (IEG)*.

Modifique el script de ejemplo para incluir estos atributos como se muestra a continuación:

```
<ieg-script  
  finish-page="IEG2_listAllIEG2Scripts"  
  quit-page="IEG2_listAllIEG2Scripts"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="ieg-schema.xsd">  
  ...  
</ieg-script>
```

Figura 56. Script con finish-page y quit-page definidos

En el ejemplo anterior, la finalización o la salida del script dará como resultado la redirección a la lista de todos los scripts IEG en las pantallas de administración.

7.4 Ejecución del reproductor IEG en una pestaña

Ejecutar el reproductor IEG en una pestaña requiere menos esfuerzo que ejecutarlo en un modal. Necesita que el enlace de 'apertura' apunte a `ieg/Screening.do` y pase en el `executionID`. `Screening.do` invoca el reproductor IEG. Los parámetros son los siguientes:

A continuación se proporciona un ejemplo de un UIM de resolución que abre el reproductor IEG en una pestaña:

```

<?xml version="1.0" encoding="UTF-8"?>
<PAGE PAGE_ID="System_IEGResolver">
  <JSP_SCRIPTLET>
    <![CDATA[

      String scriptID = request.getParameter("scriptID");
      String scriptType = request.getParameter("scriptType");
      String scriptVersion = request.getParameter(
        "scriptVersion");
      String schemaName = request.getParameter("schemaName");
      String name = request.getParameter("name");

      String executionIDParam =
        request.getParameter("executionIDParam");
      String url = null;

      curam.omega3.request.RequestHandler
        rh = curam.omega3.request.
      RequestHandlerFactory.getRequestHandler(request);

      String context = request.getContextPath() + "/";

      if (executionIDParam == null) {
        // Es necesario comprobar si hay errores de validación de script
        // antes de ejecutar el script.

        String contextWithUserPreferences = context +
          curam.omega3.user.UserPreferencesFactory
            .getUserPreferences(
              pageContext.getSession()).getLocale() + "/";

        curam.interfaces.IEGScriptAdminPkg.
        IEGScriptAdmin_checkForScriptErrors_TH
          iegScriptAdminCheckForErrors
            = new curam.interfaces.IEGScriptAdminPkg.
        IEGScriptAdmin_checkForScriptErrors_TH();

        iegScriptAdminCheckForErrors.setFieldValue(
          iegScriptAdminCheckForErrors.key$scriptID_idx, scriptID);
        iegScriptAdminCheckForErrors.setFieldValue(
          iegScriptAdminCheckForErrors.key$scriptType_idx,
            scriptType);
        iegScriptAdminCheckForErrors.setFieldValue(
          iegScriptAdminCheckForErrors.key$scriptVersion_idx,
            scriptVersion);
        iegScriptAdminCheckForErrors.setFieldValue(
          iegScriptAdminCheckForErrors.key$schemaName_idx,
            schemaName);
        //Llamar al método.
        iegScriptAdminCheckForErrors.callServer();

        String errorsPresentInScript =
          iegScriptAdminCheckForErrors.getFieldValue(
            iegScriptAdminCheckForErrors
              .result$errorsExist_idx);
        boolean errorsPresent =
          Boolean.valueOf(errorsPresentInScript).booleanValue();

        if (errorsPresent) {

          // Si hay errores, redirigir a la página de errores
          // de validación.
          String redirectTo = contextWithUserPreferences
            + "System_listValidationErrorsForRunPage.do"
            + "?name=" + name
            + "&scriptID=" + scriptID
            + "&scriptType=" + scriptType
            + "&scriptVersion=" + scriptVersion
            + "&schemaName=" + schemaName;
          url = redirectTo + "&" + rh.getSystemParameters();

        } else {

```

7.5 Ejecución del reproductor IEG en un diálogo modal

El reproductor IEG se puede abrir en un diálogo modal y existen algunas consideraciones específicas que un desarrollador de scripts debe explicar para pertenecer al mismo.

7.5.1 Apertura del reproductor IEG en un diálogo modal

Para abrir el reproductor IEG en un diálogo modal, abra `Screening.do`, en el modal, pasando los parámetros de sistema y `executionID`, utilizando un UIM de resolución. Para realizar este proceso, se proporciona `System_IEGResolverModal.uim` listo para utilizarse:

```
<PAGE PAGE_ID="System_IEGResolverModal">
  <JSP_SCRIPTLET>
    <![CDATA[

      String scriptID = request.getParameter("scriptID");
      String scriptType = request.getParameter("scriptType");
      String scriptVersion =
        request.getParameter("scriptVersion");
      String schemaName = request.getParameter("schemaName");
      String name = request.getParameter("name");

      // Es necesario comprobar si hay errores de validación de script
      // antes de ejecutar el script.
      curam.omega3.request.RequestHandler
        rh = curam.omega3.request.
          RequestHandlerFactory.getRequestHandler(request);

      String context = request.getContextPath() + "/";
      String contextWithUserPreferences = context +
        curam.omega3.user.UserPreferencesFactory
          .getUserPreferences(
            pageContext.getSession()).getLocale() + "/";

      String url = null;

      curam.interfaces.IEGScriptAdminPkg.
        IEGScriptAdmin_checkForScriptErrors_TH
        iegScriptAdminCheckForErrors
        = new curam.interfaces.IEGScriptAdminPkg.
          IEGScriptAdmin_checkForScriptErrors_TH();

      iegScriptAdminCheckForErrors.setFieldValue(
        iegScriptAdminCheckForErrors.key$scriptID_idx,
        scriptID);
      iegScriptAdminCheckForErrors.setFieldValue(
        iegScriptAdminCheckForErrors.key$scriptType_idx,
        scriptType);
      iegScriptAdminCheckForErrors.setFieldValue(
        iegScriptAdminCheckForErrors.key$scriptVersion_idx,
        scriptVersion);
      iegScriptAdminCheckForErrors.setFieldValue(
        iegScriptAdminCheckForErrors.key$schemaName_idx,
        schemaName);
      //Llamar al método.
      iegScriptAdminCheckForErrors.callServer();

      String errorsPresentInScript =
        iegScriptAdminCheckForErrors.getFieldValue(
          iegScriptAdminCheckForErrors.result$errorsExist_idx);
      boolean errorsPresent =
        Boolean.valueOf(errorsPresentInScript).
          booleanValue();

    ]]>
  </JSP_SCRIPTLET>
</PAGE>
```

```

if (errorsPresent) {

    // Si hay errores, redirigir a la
    // página de errores de validación.
    String redirectTo = contextWithUserPreferences
        + "System_listValidationErrorsForModalPage.do"
        + "?name=" + name + "&scriptID=" + scriptID
            + "&scriptType=" + scriptType
        + "&scriptVersion=" + scriptVersion
        + "&schemaName=" + schemaName;
    url = redirectTo + "&" + rh.getSystemParameters();

} else {

    // Llamar al método de ejecución de script y redirigirlo
    // al reproductor IEG.
    curam.interfaces.IEGScriptAdminPkg.
        IEGScriptAdmin_runScript_TH iegScriptAdminRunScript
        = new curam.interfaces.IEGScriptAdminPkg.
            IEGScriptAdmin_runScript_TH();

    iegScriptAdminRunScript.setFieldValue(
        iegScriptAdminRunScript.key$dtls$scriptID_idx,
        scriptID);
    iegScriptAdminRunScript.setFieldValue(
        iegScriptAdminRunScript.key$dtls$scriptType_idx,
        scriptType);
    iegScriptAdminRunScript.setFieldValue(
        iegScriptAdminRunScript.key$dtls$scriptVersion_idx,
        scriptVersion);
    iegScriptAdminRunScript.setFieldValue(
        iegScriptAdminRunScript.key$schemaName_idx,
        schemaName);
    //Llamar al método.
    iegScriptAdminRunScript.callServer();

    String executionID =
        iegScriptAdminRunScript.getFieldValue(
            iegScriptAdminRunScript.result$executionID_idx);
    executionID = executionID.replaceAll(",", "");

    url = context + "ieg/Screening.do?"
        + "executionID=" + executionID
        + "&" + rh.getSystemParameters();
}

// Redirigir a la página correcta.
response.sendRedirect(
    response.encodeRedirectURL(url));
]]>
</JSP_SCRIPTLET>
</PAGE>

```

7.5.2 Salida de una ejecución de script en un diálogo modal

Hay dos enfoques generales que un desarrollador de script puede adoptar para completar una ejecución de script IEG en un diálogo modal o salir de dicha ejecución:

- Cerrar directamente el diálogo modal y renovar o redirigir en la pestaña padre.
- Progresar a la(s) pantalla(s) UIM adicional(es) en el diálogo modal.

7.5.2.1 Cierre directo del modal al finalizar script

Para cerrar un diálogo modal directamente al finalizar o salir (acciones Salir, Guardar y salir) de una ejecución de script IEG, el desarrollador de script debe especificar un UIM de resolución como página de finalización y/o página de salida. Ese UIM de resolución debe a su vez invocar una JSP personalizada que llame a la función JavaScript adecuada para cerrar el diálogo.

Por ejemplo, para redirigir a la pantalla de administración IEG2_listAllIEG2Scripts, incluya el siguiente scriptlet JSP en el archivo UIM:

```
<PAGE
  PAGE_ID="IEG2_resolveFinishScript"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://Curam/UIMSchema.xsd"
>
<JSP_SCRIPTLET>
<![CDATA[

  curam.omega3.request.RequestHandler
  rh = curam.omega3.request.RequestHandlerFactory
    .getRequestHandler(request);

  String context = request.getContextPath() + "/";
  context += curam.omega3.user.UserPreferencesFactory
    .getUserPreferences(
      pageContext.getSession()).getLocale() + "/";

  String url = "";
  url = context + "IEG2_listAllIEG2ScriptsPage.do";

  String forwardParams =
    request.getParameter("forwardParams");

  if (screenContext != null && screenContext
    .hasContextBits(
      curam.omega3.taglib.ScreenContext.MODAL)) {
    url += "?" + rh.getSystemParameters();
    String encodeRedirectURL = response.encodeURL(url);
    response.sendRedirect(response.encodeRedirectURL(
      request.getContextPath() +
        "/ieg/CloseAndRedirect.jsp?redirect="
        + encodeRedirectURL));
  } else {
    url += "?" + rh.getSystemParameters();
    response.sendRedirect(
      response.encodeRedirectURL(url));
  }

  ]]>
</JSP_SCRIPTLET>
</PAGE>
```

CloseAndRedirect.jsp se proporciona listo para cerrar el diálogo modal y redirigir a un UIM especificado (si se proporciona) en el padre.

7.5.2.2 Progresión a pantallas UIM adicionales en el diálogo modal

Para mantener el diálogo modal abierto para visualizar pantallas UIM adicionales después de que se haya completado la ejecución de script, especifique la página UIM necesaria como página de finalización y/o página de salida en la definición de script IEG. Una vez que haya cargado UIM, habrá salido de IEG y se aplicará el proceso UIM estándar en un diálogo modal.

7.6 Limpieza de datos de aplicación

La limpieza de datos de aplicación implica la eliminación de datos de la tabla de base de datos IEGEXECUTIONSTATE y del almacén de datos (DS), donde sea apropiado. En esta sección se detallan las tareas de limpieza de datos manuales y automáticas que los autores de script deben conocer y se hacen algunas recomendaciones para garantizar que la limpieza de datos de aplicación puede continuar sin contratiempos.

Con el fin de soportar la ejecución de un script IEG, el motor IEG debe mantener información sobre las ejecuciones de script individuales. Por ejemplo, el motor IEG debe hacer un seguimiento de la página actual para la ejecución de script. El motor IEG también debe mantener una lista de las páginas que se han presentado al usuario para dar soporte a la navegación. Las respuestas a las preguntas de control no son persistentes en el DS y el motor IEG también debe hacer un seguimiento de las mismas. Toda la información para soportar la ejecución de un script IEG se conserva en la tabla IEGEXECUTIONSTATE. Cuando se crea un objeto IEGScriptExecution nuevo a través de la API IEGScriptExecutionFactory, se crea una entrada correspondiente en la tabla IEGEXECUTIONSTATE. La tabla IEGEXECUTIONSTATE es una tabla "interna" destinada sólo a ser utilizada por el motor IEG y no se debe modificar o ampliar. Los autores de script no deben hacerse dependientes o realizar suposiciones sobre el contenido de esta tabla puesto que pueden estar sujetos a cambios sin previo aviso.

IEG no tiene forma de saber si una entrada en la tabla IEGEXECUTIONSTATE ya no es necesaria y, por lo tanto, las entradas se mantendrán hasta que se supriman explícitamente. Para evitar que la tabla IEGEXECUTIONSTATE se recargue de forma innecesaria, si una ejecución de script se ha completado o no se reanudará o volverá a ejecutar, la entrada de la tabla se debe eliminar a través del método `removeScriptExecutionObject` de la API `IEGScriptExecutionFactory`.

IEG no puede hacer ninguna deducción sobre qué datos se pueden utilizar para identificar de forma lógica y exclusiva una determinada ejecución de script porque esto puede variar de una definición de script a otra. La única manera para que IEG identifique una ejecución de script es a través del ID generado que se asigna a la ejecución de script cuando se crea inicialmente. Es muy recomendable que los autores de script implementen un mecanismo para identificar las ejecuciones de script asociando datos exclusivos con los ID de ejecución de script. Se puede crear una nueva tabla para mantener la relación entre los datos que identifican la ejecución y el ID de ejecución para facilitar la reanudación de las ejecuciones de script. Cuando ya no son necesarios se pueden eliminar. La eliminación de una ejecución de script no hace que se elimine ninguno de los datos recopilados que se conservan en el DS.

De forma similar a IEGEXECUTIONSTATE, el motor IEG y el DS no tienen manera de saber cuándo ya no son necesarios los datos que se recopilan durante una ejecución de script y se conservan en el DS. De nuevo, el DS se puede recargar de forma innecesaria con entidades que ya no son necesarias. Está previsto que las entidades no permanezcan en el DS indefinidamente, sino que los datos recopilados se muevan a las tablas de aplicación y, a continuación, se eliminen del DS. Cuando se suprime una entidad del DS, también se suprimen las entidades hijo. Por lo tanto, cuando los datos que se recopilan durante una ejecución de script se han trasladado a las tablas de aplicación y ya no se necesitan, es suficiente suprimir la entidad raíz para la ejecución.

El fragmento de código de ejemplo siguiente muestra la supresión de la entidad raíz:

```
final Long applicationID = execution.getRootEntityID();
    final Entity rootEntity = datastore.readEntity(applicationID);
    rootEntity.delete();
```

Figura 58. Supresión de la entidad raíz

7.7 Reanudación de scripts ejecutados

Es posible detener una ejecución del script y reanudarla posteriormente. Para ello, la aplicación debe tener cuidado de almacenar el ID de ejecución en una tabla personalizada y asociarlo con algún ID de usuario. Consulte 7.6, "Limpieza de datos de aplicación", en la página 52 para obtener más detalles.

Siempre que la tabla IEGEXECUTIONSTATE no se haya limpiado y la definición de script no se haya modificado, la ejecución de script se reanudará pasando el parámetro `executionID` al reproductor IEG de la misma manera que se realiza cuando se inicia una nueva ejecución de script.

Capítulo 8. Gestión de datos capturados

8.1 Introducción

Como se ha mencionado anteriormente, los datos capturados durante la ejecución de script se conserva en el almacén de datos (DS). En este capítulo se explicará cómo puede recuperar los datos capturados del DS. En este capítulo también se explicará cómo se pueden insertar datos en el DS para que estén disponible para IEG mientras se ejecutan scripts.

8.2 Recuperación de datos capturados

El almacén de datos (DS) tiene una API pública que se puede utilizar en el código de aplicación. Esta API se utiliza muy frecuentemente para recuperar información de un esquema lleno pero también puede utilizarse para rellenar previamente un esquema. Por ejemplo, una vez que un cliente ha completado una aplicación, puede enviar la información. En este punto, la API se puede utilizar para extraer los datos del esquema y llenar las tablas en la base de datos relacional.

Un llenado previo se produce, por ejemplo, cuando se conoce información sobre el cliente antes de iniciar la aplicación. Si se necesita parte de esa información navegar por la aplicación, el DS se puede llenar previamente con la información.

Para leer los datos de un esquema, es necesario conocer la ejecución adecuada del script. Esto significa que se está recuperando la información de aplicación correcta para un cliente. Por lo tanto, el ID de ejecución y el nombre de esquema son de vital importancia para poder acceder a los datos.

El fragmento de código de ejemplo siguiente muestra la obtención de la entidad raíz:

```
final IEGRuntime runtimeAPI = new IEGRuntime();
final IEGRootEntityID rootEntityID =
    runtimeAPI.getScriptExecutionRootEntityID(executionID);

Datastore ds = DatastoreFactory.newInstance()
    .openDatastore(kSchemaName);

final Entity rootEntity =
    ds.readEntity(rootEntityID.entityID);
```

Figura 59. Obtención de entidad raíz

Desde aquí, la entidad raíz puede utilizarse para recuperar otras entidades bajo esta entidad raíz.

8.3 Llenado previo de scripts con datos capturados

Es posible llenar previamente los valores que se mostrarán al usuario para que las respuestas sólo se deban confirmar o modificar.

Por ejemplo, se puede llenar previamente el nombre y la fecha de nacimiento de un usuario en una página de Detalles personales suponiendo que el usuario ya haya iniciado la sesión y otra base de datos contenga los detalles personales.

El DS se puede llenar previamente antes del inicio de la ejecución de script de la manera siguiente:

```

...
Datastore ds = null;

try {
    // abrir el almacén de datos y crear la entidad raíz
    ds = DatastoreFactory.newInstance().openDatastore(schemaName);
} catch (NoSuchSchemaException e) {
    throw new AppException(IEG.ID_SCHEMA_NOT_FOUND);
}

final EntityType appType = ds.getEntityType("Application");
final Entity rootElement = ds.newEntity(appType);

ds.addRootEntity(rootElement);

final EntityType personType = ds.getEntityType("Person");
final Entity person = ds.newEntity(personType);

person.setAttribute("firstName", "TestFirstName");
person.setAttribute("lastName", "TestLastName");
person.setAttribute("dateOfBirth", "19700101");
//...

rootElement.addChildEntity(person);

```

Figura 60. Fragmento de código que llena el DS

Entonces la entidad raíz se puede utilizar en la creación de una nueva ejecución de script de la siguiente manera:

```

...

// crear la ejecución de script
final IEGRootEntityID rootEntityID = new IEGRootEntityID();
rootEntityID = rootElement.getUniqueID();
final IEGRuntime runtimeAPI = new IEGRuntime();
final IEGScriptExecutionIdentifier executionIdentifier =
    runtimeAPI.createScriptExecutionExistingRootEntity(
        iegScriptID, schemaName, rootEntityID);

```

Figura 61. Creación de una ejecución de script

Entonces se puede ejecutar el reproductor IEG utilizando esta nueva ejecución de script de la manera siguiente:

```

<?xml version="1.0" encoding="UTF-8"?>
<PAGE PAGE_ID="IEGScriptLauncher">
  <JSP_SCRIPTLET>
    <![CDATA[
curam.omega3.request.RequestHandler rh =
  curam.omega3.request.RequestHandlerFactory.getRequestHandler(
    request);

String context = request.getContextPath() + "/";

String url =
  context + "ieg/Screening.do?" + "executionID=" + executionID
  + "&" + rh.getSystemParameters();

// Redirigir a la página correcta.
response.sendRedirect(response.encodeRedirectURL(url));
]]>
  </JSP_SCRIPTLET>
</PAGE>

```

Figura 62. Inicio del reproductor IEG

Tenga en cuenta que sólo se puede llenar previamente el DS y no las preguntas de control u otra información relacionada con el script ya que éstas se almacenan en la ejecución del script y no en el DS. Esto significa que no es posible llenar previamente los datos visualizados en la primera sección del script y empezar en la segunda sección. La primera sección se visualizará y el usuario podrá confirmar los datos llenados previamente.

Capítulo 9. Utilización del almacén de recursos

9.1 Introducción

El almacén de recursos es un área de la base de datos de infraestructura que se utiliza para almacenar recursos utilizados en una aplicación activa. Los recursos pueden ser de cualquier tipo, pero los que IEG utiliza más comúnmente son los recursos de archivo de propiedades e imágenes.

9.2 Listado de todos los recursos

Para obtener acceso a las pantallas de administración de recursos, necesitará iniciar la sesión como usuario administrativo. Una vez que haya iniciado la sesión, verá una sección en el panel de navegación denominada IEG. Al pulsar en la sección, verá un menú que contiene un enlace denominado 'Recursos de la aplicación'. Si pulsa en él, se visualizará una lista de recursos con un recuadro de búsqueda basado en la categoría.

Los recursos se organizan en categorías. Los recursos existentes se visualizan seleccionando una categoría en los criterios de filtro y seleccionando 'buscar en'. Las categorías de recursos utilizadas por IEG son las siguientes:

- CSS
Plantillas de hoja de estilo que se pueden modificar para personalizar el aspecto del reproductor IEG.
- Imagen
Imágenes utilizadas en el reproductor IEG y los scripts IEG.
- Propiedad
Archivos de propiedades que contienen texto específico de la configuración regional para Scripts y Páginas de preguntas.

9.3 Carga de un nuevo recurso

En la parte superior de la pantalla que lista todos los recursos hay un enlace que le permite añadir un recurso nuevo. Al pulsar este enlace, se presenta una pantalla donde se deben entrar los detalles de recurso.

Debe especificar la siguiente información:

- Nombre
Es un nombre exclusivo para el recurso que se puede utilizar dentro de un script IEG para hacer referencia al mismo. En función del tipo de recurso, se puede aplicar una convención de denominación para utilizarla dentro de un script IEG. Las secciones que tratan sobre la 9.7, "Adición de imágenes", en la página 60 y el 9.8, "Cambio de texto estático", en la página 60 contienen más detalles.
- Tipo de contenido
Al entregar un recurso a un navegador web, se necesita un tipo de contenido para indicar al navegador cómo gestionar el recurso. Los tipos de contenido más comunes utilizados en un script IEG serán `image/png` para una imagen PNG y `text/plain` para un archivo de propiedades.
- Contenido
El selector de archivos permite al usuario elegir el recurso que se debe cargar.

La información siguiente es opcional:

- Categoría
La categoría en la que se debe añadir el nuevo recurso.

- Disposición del contenido
Para los recursos utilizados en los scripts IEG, puede dejarse vacío.
- Entorno local
Si desea tener una versión de un recurso específica de la configuración regional, entre aquí el código de configuración regional. Cuando el sistema busca un recurso, utiliza un mecanismo de emergencia parecido a Java. Por ejemplo, si la configuración regional actual es en_US el sistema intentará localizar el recurso para la configuración regional en_US, a continuación en y finalmente el recurso “predeterminado”. El recurso “predeterminado” se especifica dejando vacío el campo de configuración regional al cargar el recurso.
- Interno
Indica si el recurso es sólo para uso interno y no debe proporcionarse nunca al navegador web. En este primer release de IEG, se puede ignorar este valor.
- Descripción
Descripción del recurso.

9.4 Eliminación de un recurso existente

Para suprimir un recurso existente, seleccione el enlace 'ver' en el recurso y en la página 'Ver recurso' seleccione 'suprimir' para eliminar este recurso del sistema. Al pulsar este enlace, se visualizará un diálogo de confirmación que le solicitará que confirme que desea eliminar este recurso del sistema.

9.5 Actualización de un recurso existente

Para actualizar un recurso existente, seleccione el enlace 'editar' en la página 'Recursos de la aplicación' o en la página 'Ver recurso'. A continuación, puede examinar el recurso actualizado en el sistema de archivos en el campo 'Nuevo contenido'.

9.6 Descarga de un recurso existente

Cada entrada de la lista de recursos se puede descargar pulsando el enlace de "descarga" en la página "Recursos de la aplicación". Este enlace abrirá el diálogo de descarga de archivo de navegador para permitir al usuario guardar el recurso o abrirlo directamente.

9.7 Adición de imágenes

Los scripts IEG le permiten especificar imágenes que se utilizarán para las secciones (de forma predeterminada en el panel de navegación a la izquierda de una página) y las páginas (en el área de título de la página) y también vienen con algunas imágenes que están incorporadas en el sistema (como las diversas imágenes de personas utilizadas en las pestañas de personas, etc.). Todas estas imágenes se deben almacenar en el almacén de recursos, para que se puedan añadir imágenes nuevas y actualizar las existentes sin tener que reconstruir y volver a desplegar la aplicación. Al subir un recurso de imagen, establezca el “Tipo de contenido” de forma adecuada para la imagen (por ejemplo, image/png, image/gif etc.) y deje el campo “Disposición del contenido” vacío.

9.8 Cambio de texto estático

El motor IEG le permite entrar todo el texto del script para la configuración regional predeterminada directamente en la definición de script. Sin embargo, éste no es el lugar en el que realmente se lee el texto que se visualiza en las pantallas. En lugar de ello, todo el texto al que se hace referencia desde dentro de un script se almacena en archivos de propiedades específicos de la configuración regional en el almacén de recursos. Para cada script, habrá un mínimo de un archivo de propiedades para el propio script y un archivo de propiedades para cada página en el script. Para garantizar la exclusividad de estos archivos, se utiliza el siguiente convenio de denominación (obviamente la última parte sólo es aplicable a los archivos de propiedades específicas de página):

IDscript_versiónScript_tipoScript_IDpágina

Cuando utilice las pantallas de administración IEG para subir un script nuevo al sistema, todo el texto estático contenido en él (por ejemplo todas las etiquetas, títulos, descripciones, etc.) se extrae automáticamente en los archivos de propiedades con los debidos nombres para el script y se almacena en el almacén de recursos sin ninguna configuración regional asociada a ellos (para que actúen como las propiedades de emergencia si no existen propiedades para la configuración regional en la que se está ejecutando). Entonces se puede cambiar cualquier parte de este texto simplemente descargando el archivo de propiedades actual, teniendo presente el convenio de denominación descrito anteriormente para localizar el recurso en la lista de recursos. A continuación, realice los cambios necesarios y actualice el recurso como se describe en 9.5, “Actualización de un recurso existente”, en la página 60. No es necesario realizar cambios en el script propiamente dicho.

Asimismo, se pueden añadir fácilmente versiones de estos archivos para otras configuraciones regionales y éstas se elegirán antes que las propiedades del configuración regional predeterminada la próxima vez que el script se ejecute en dicha configuración regional. Al cargar un recurso de archivo de propiedades, establezca el “Tipo de contenido” en texto/sin formato (`text/plain`) y deje el campo “Disposición del contenido” vacío.

9.9 Cambio de la codificación de archivo predeterminada

Al cargar un recurso de texto sin formato, se espera que el archivo esté en la codificación UTF-8. Si desea utilizar una codificación diferente al subir el archivo, se puede utilizar el campo "Tipo de contenido" para especificarlo mediante el uso del parámetro opcional `charset`. Por ejemplo:

```
text/plain; charset=ISO-8859-1
```

Capítulo 10. Utilización de IBM Rational AppScan para explorar IEG

10.1 Introducción

En este capítulo se describen los pasos necesarios para realizar exploraciones de seguridad de aplicaciones de estilo IEG utilizando la herramienta IBM®Rational AppScan herramienta.

10.2 Preparación

En IEG la comunicación entre el reproductor y el motor se coordina por medio de una señal de sincronización. La señal de sincronización se utiliza para asegurarse de que la página enviada por el navegador es coherente con la página que el motor IEG está esperando que se envíe. Esto facilita la detección cuando el usuario utiliza los botones de navegación del navegador en lugar de los botones de navegación en el propio reproductor. La señal de sincronización cambia para cada página de pregunta que se visualiza en el reproductor IEG. Esto hace que sea muy difícil explorar los scripts de preguntas de IEG que se ejecutan en el reproductor IEG.

Por este motivo, antes de la exploración, se recomienda establecer la propiedad de configuración de script `appscan.mode.enabled` en `true`. Cuando esta propiedad está establecida en `true`, el motor no comprueba el valor o la señal de sincronización pasada por el reproductor. La inhabilitación de la comprobación de señal de sincronización es aceptable cuando se realiza una exploración, pero la comprobación de señal de sincronización deberá estar siempre habilitada en un entorno de producción.

Además, a fin de reducir la cantidad de información superflua de la que se informa en una exploración, el rastreo de pila debe inhabilitarse. Para inhabilitar el rastreo de pila:

- Vaya a la carpeta `webclient\JavaSource\curam\omega3\`
- Cambie el nombre de `Initial_ApplicationConfiguration.properties` por `ApplicationConfiguration.properties`
- Abra `ApplicationConfiguration.properties`
- Añada la entrada: `errorpage.stacktrace.output=false`

10.3 Páginas de relaciones

Las páginas de relaciones son una característica especial de IEG que facilita la recopilación de información acerca de las relaciones entre las personas de una unidad familiar. A diferencia de las demás páginas de un script IEG, las páginas de relaciones tienen una naturaleza más dinámica y contienen un número variable de campos. Actualmente los nombres de los campos que se generan para las páginas de relaciones varían de una ejecución a otra. Esto significa que actualmente no es posible ejecutar una exploración en un script de preguntas IEG que contiene una página de relaciones.

10.4 Configuración de la exploración

Una vez que se ha iniciado AppScan, se puede crear una exploración nueva seleccionando la opción 'Create New Scan...' (Crear exploración nueva...) en la pantalla de bienvenida.

A continuación, seleccione 'Regular Scan' (Exploración regular) en las plantillas predefinidas en la pantalla siguiente.

Elija 'Web Application Scan' (Exploración de aplicación web) en la primera página del asistente de configuración y pulse 'Siguiente'.

En la página 'URL and Servers' (URL y servidores) del asistente, entre el URL de inicio de la aplicación. Una vez que se ha entrado el URL, se puede verificar pulsando el icono situado junto al campo de entrada. Esto hará que el navegador AppScan se visualice y se intentará abrir el URL. Confirme que el URL es correcto y se puede acceder a él. (Pulse yes (sí) en el aviso de seguridad si es necesario). Cierre el navegador y pulse 'Siguiente' en el asistente de configuración.

Entre los detalles de gestión de inicio de sesión necesarios. Las aplicaciones que se ejecutan bajo Eclipse/Tomcat no requieren que el usuario inicie la sesión, por lo tanto se puede seleccionar la opción 'Ninguna'. Pulse 'Siguiente'.

En la pantalla 'Test Policy' (Política de pruebas) pulse el enlace 'Full Scan Configuration' (Configuración de exploración completa) en el panel 'General Tasks' (Tareas generales). Esto presenta el diálogo 'Scan Configuration' (Configuración de la exploración).

10.5 Política de pruebas

Asegúrese de que 'Política de pruebas' se ha seleccionado en el panel de selección de vista en el lado izquierdo del diálogo de configuración. El enfoque más directo al configurar una exploración consiste en permitir todas las pruebas y luego inhabilitar las pruebas de valor bajo que aumentan el tiempo necesario para ejecutar la exploración.

Seleccione 'Habilitado/Inhabilitado' en el menú desplegable 'clasificar pruebas por'. Primero compruebe los recuadros 'Parcialmente habilitados' y, a continuación, los 'Inhabilitados'. La única entrada visualizada debe ser 'Habilitado'. Seleccione 'Gravedad' en el menú desplegable. Quite la marca de selección de los recuadros 'Bajo' e 'Informativo'. Para realizar la exploración de IEG no es necesario realizar pruebas invasivas ya que estas pruebas están más relacionadas con la prueba de la plataforma. Seleccione 'Invasiva' en el menú desplegable. Quite la marca de selección del recuadro 'Invasiva'.

10.6 Opciones de exploración

Seleccione 'Opciones de exploración' en el panel de selección de vista. Establezca el límite de vía de acceso redundante ('Redundant Path Limit') en 1. Elija la amplitud primero ('Breadth First') como el 'Método de exploración'.

10.7 Comunicaciones y proxy

Seleccione 'Comunicaciones y proxy' en el panel de selección de vista. Establezca el número de hebras ('Number of Threads') en 1.

10.8 Opciones de prueba

Seleccione 'Opciones de prueba' en el panel de selección de vista. Quite la marca de selección de 'Use Adaptive Testing based on application behavior' (Utilizar prueba adaptativa basándose en el comportamiento de aplicación).

10.9 Operaciones de varios pasos

IEG requiere que se utilicen datos formateados correctamente en determinados parámetros. Como tal, AppScan se debe 'formar' para utilizar la aplicación que se está probando. Seleccione 'Operaciones de varios pasos' en el panel de selección de vista. Pulse el botón de registro. Esto hará que se visualice el navegador AppScan e intentará abrir el URL especificado en la página 'URL y servidores' del asistente de configuración. A continuación, debe navegar en la aplicación según sea necesario, entrando datos relevantes. AppScan registrará los valores entrados y utilizará estos valores para cada prueba que ejecute posteriormente. Cuando haya terminado, simplemente cierre el navegador. El diálogo de configuración de exploración se actualizará con la secuencia que se acaba de registrar. Marque la casilla de verificación

para habilitar la reproducción de esta secuencia ('Enable playback of this sequence') y quite la marca de la casilla de verificación para permitir la optimización de reproducción ('Allow play optimization').

Tome nota de todos los pasos de secuencia que contienen Screening.do. Tendrá que convertir estos pasos de secuencia en expresiones regulares y añadirlos como vías de excepción a las opciones de vías de exclusión de AppScan. AppScan puede quedar fácilmente fuera de sincronización cuando se trata de operaciones registradas, de modo que debe asegurarse de que AppScan ignorará la vía incorrecta y se mantendrá en el script registrado cuando se ejecuten las pruebas. Esto se consigue indicando a AppScan que ignore todos los pasos de secuencia que contienen screening.do, excepto los que se especifiquen en expresiones regulares. Tome nota de cada valor `__u=x` encontrado en la lista de pasos de secuencia.

10.10 Excluir vías de acceso y archivos

Seleccione 'Excluir vías de acceso y archivos' en el panel de selección de vista. Pulse el botón para añadir una vía de acceso de exclusión. Elija 'Excluir' como el 'Tipo' y seleccione 'Expresión regular' en el menú desplegable 'Coincidencia'. Entre `./Curam/ieg/Screening.do.*` para la 'Vía de acceso' y pulse 'Aceptar'.

Añada otra vía de acceso de exclusión. Elija 'Excepción' como el 'Tipo' y seleccione 'Expresión regular' en el menú desplegable 'Coincidencia'. Entre `./Curam/ieg/Screening.do?executionID=.\\d*` para la 'Vía de acceso' y pulse 'Aceptar'.

También se deberá añadir una excepción para cada valor `__u=x` encontrado en la lista de pasos de secuencia. De nuevo, seleccione 'Expresión regular' en el menú desplegable el 'Coincidencia' y entre una expresión en el siguiente formato para 'Vía de acceso': `./Curam/ieg/Screening.do?executionID=.* &__u=[valor mostrado en pantalla de resumen]`

Pulse 'Aceptar'.

Pulse 'Aceptar' para volver al asistente de configuración.

Pulse 'Siguiente' en el asistente de configuración.

10.11 Completo

En este punto, la configuración de la exploración se ha completado. Elija la opción 'I will start my scan later' (Empezaré mi exploración más tarde) para que se guarde la exploración configurada en lugar de permitir que AppScan explore la aplicación entera de forma aleatoria. Pulse "Finalizar".

10.12 Ejecución de la exploración

Para iniciar la exploración, seleccione el elemento de menú de exploración ('Scan') en la ventana principal de AppScan y seleccione 'Test Multi-Step Operations Only' (Probar sólo operaciones de varios pasos). En función de la aplicación que se deba probar, una exploración puede tardar varios días en completarse. Una vez que la exploración se haya completado, AppScan mostrará los resultados de la exploración en una pantalla de resumen. A continuación, se deben investigar estos resultados para determinar qué problemas indicados son vulnerabilidades reales y cuáles son falsos positivos.

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. Solicite información al representante local de IBM acerca de los productos y servicios disponibles actualmente en su zona. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implica que sólo pueda utilizarse ese producto, programa o servicio de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ningún derecho de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM. IBM podría tener patentes o solicitudes de patentes pendientes relacionadas con el tema principal que se describe en este documento. La posesión de este documento no confiere 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

EE.UU.

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

Intellectual Property Licensing

Legal and Intellectual Property Law.

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokio 103-8510, Japón

El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde las disposiciones en él expuestas sean incompatibles con la legislación local: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍA DE NINGUNA CLASE, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERABILIDAD, COMERCIALIZACIÓN O IDONEIDAD PARA UN PROPÓSITO DETERMINADO. Algunos países no permiten la renuncia a garantías explícitas o implícitas en determinadas transacciones, por lo que puede que esta declaración no sea aplicable en su caso.

La información de este documento puede incluir imprecisiones técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; estos cambios se incorporarán en nuevas ediciones de la publicación. IBM puede reservarse el derecho de realizar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento sin previo aviso.

Cualquier referencia incluida en esta información a sitios web que no sean de IBM sólo se proporciona para su comodidad y en ningún modo constituye una aprobación de dichos sitios web. El material de esos sitios web no forma parte del material de este producto de IBM y la utilización de esos sitios web se realizará bajo su total responsabilidad.

IBM puede utilizar o distribuir cualquier información que se le facilite de la manera que considere adecuada, sin contraer por ello ninguna obligación con el remitente. Los titulares de licencias de este programa que deseen obtener información sobre el mismo con el fin de permitir: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido éste) y el uso mutuo de información que se haya intercambiado, deben ponerse en contacto con:

IBM Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

EE.UU.

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

IBM proporciona el programa bajo licencia que se describe en este documento y todo el material bajo licencia disponible para el mismo bajo los términos del Acuerdo de cliente de IBM, el Acuerdo internacional de licencias de programas de IBM o cualquier acuerdo equivalente entre las partes.

Los datos de rendimiento incluidos aquí se determinaron en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar considerablemente. Algunas mediciones podrían haberse realizado en sistemas en desarrollo y, por lo tanto, no existe ningún tipo de garantía de que dichas mediciones sean las mismas en los sistemas con disponibilidad general. Además, es posible que algunas mediciones se hayan calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a sus entornos específicos.

La información relacionada con productos que no son de IBM se ha obtenido de los proveedores de dichos productos, de sus anuncios publicados o de otras fuentes de disponibilidad pública.

IBM no ha probado estos productos y no puede confirmar la precisión de rendimiento, compatibilidad ni otras afirmaciones relacionadas con productos que no son de IBM. Las preguntas relativas a las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de dichos productos.

Las afirmaciones relativas a las intenciones futuras de IBM están sujetas a cambio o retirada sin previo aviso, y sólo representan objetivos

Todos los precios de IBM que se muestran son precios de distribuidor recomendados por IBM, corresponden al momento actual y están sujetos a cambios sin aviso previo. Los precios de los distribuidores pueden variar.

Esta información se ofrece con fines de planificación únicamente. La información incluida en este documento puede cambiar antes de que los productos descritos estén disponibles.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlos de la manera más completa posible, los ejemplos incluyen los nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con nombres y direcciones utilizados por empresas comerciales reales son mera coincidencia.

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 exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede garantizar ni implicar la fiabilidad, capacidad de servicio o función de estos programas. Los programas de ejemplo se proporcionan "TAL CUAL", sin garantía de ningún tipo. IBM no es responsable de ningún daño resultante de la utilización de los programas de ejemplo por parte del usuario.

Todas las copias o fragmentos de las copias de estos programas de ejemplo o cualquier trabajo que de ellos se derive, deberán incluir un aviso de copyright como el que se indica a continuación:

© (el nombre de la empresa) (año). Algunas partes de este código proceden de los programas de ejemplo de IBM Corp.

© Copyright IBM Corp. _escriba el año o los 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.

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>.

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

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