



Transformaciones

gvSIG: Avanzando Juntos

<http://www.gvsig.gva.es>
<http://www.gvsig.org>

Jorge Piera Llodrá
Iver T.I.
jpiera@gvsig.org

Índice de la Presentación

1. Definición de transformación.

1.1. Transformaciones a nivel de DAL.

1.2. Transformaciones a nivel de aplicación.

2. El wizard de las transformaciones.

2.1. Uso del wizard.

2.2. Creación de un nuevo wizard.



Índice de la Presentación

3.Las transformaciones.

3.1. Creación de una transformación.

3.2. Los panelas asociados a la transformación.



Índice de la Presentación

1. Definición de transformación.

1.1. Transformaciones a nivel de DAL.

1.2. Transformaciones a nivel de aplicación.

2. El wizard de las transformaciones.

2.1. Uso del wizard.

2.2. Creación de un nuevo wizard.

Definición de Transformación



Definición de Transformación

COL1	COL2	COL3



COL1	COL2	COL3	COL4

Definición de Transformación

El tema de eventos

El tema de eventos transforma un origen de datos con dos columnas con las coordenadas X e Y en un origen de datos con una geometría de tipo Point.

X	Y	ATT
1.0	1.0	feli
2.0	3.0	zcum
7.0	7.0	ple
8.0	3.0	sus
8.0	9.0	ana



X	Y	ATT	Geometry
1.0	1.0	feli	Point(1.0, 1.0)
2.0	3.0	zcum	Point(2.0, 3.0)
7.0	7.0	ple	Point(7.0, 7.0)
8.0	3.0	sus	Point(8.0, 3.0)
8.0	9.0	ana	Point(8.0, 9.0)



Índice de la Presentación

1. Definición de transformación.

1.1. Transformaciones a nivel de DAL.

1.2. Transformaciones a nivel de aplicación.

2. El wizard de las transformaciones.

2.1. Uso del wizard.

2.2. Creación de un nuevo wizard.



Transformaciones a nivel de DAL

- Las transformaciones se aplican a nivel de **DataStore** (aunque de momento, sólo son utilizables a nivel de **FeatureStore**).
- Se puede aplicar más de una transformación sobre un mismo almacén de datos.
- Lo que se hace es “crear una vista del almacén”, sin que sea posible modificarlo.

Transformaciones a nivel de DAL

- La interfaz **FeatureStoreTransform** es la que deben implementar todas las transformaciones.
- Se aplica a nivel de **FeatureStore**.
- Una transformación deberá básicamente hacer dos cosas:
 1. Definir el **FeatureType** de salida.
 2. Aplicar la transformación a una **Feature** concreta.

Transformaciones a nivel de DAL

Se aplican a nivel de **Feature**

- El proceso de transformaciones iterará sobre el **FeatureStore** y aplicará la transformación Feature a Feature.





Transformaciones a nivel de DAL

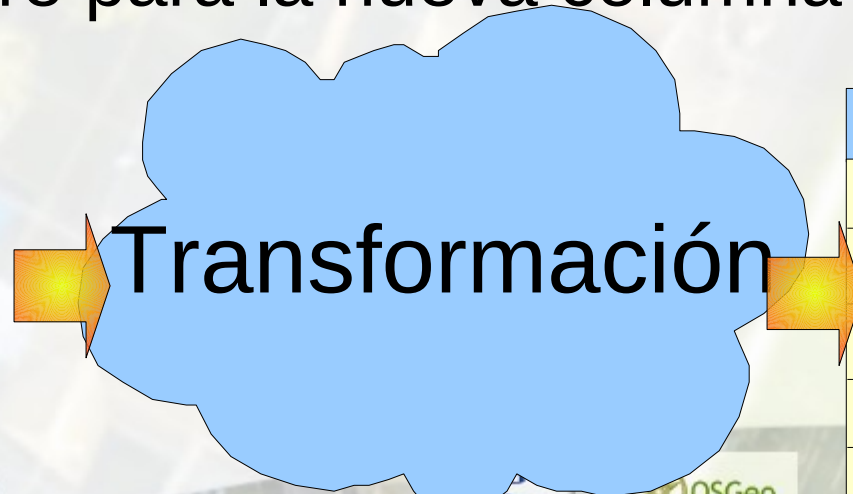
- El API de **FeatureStoreTransform** sirve para poder aplicar la transformación de forma genérica.
- Cada transformación particular se tiene que inicializar con los parámetros que necesite.

Transformaciones a nivel de DAL

Tema de eventos: **EventThemeTransform**

- Se necesitan 3 parámetros para generar un tema de eventos:
 - 1.Nombre del campo que contiene la X.
 - 2.Nombre del campo que contiene la Y.
 - 3.Nombre para la nueva columna (opcional).

X	Y	ATT
1.0	1.0	feli
2.0	3.0	zcum
7.0	7.0	ple
8.0	3.0	sus
8.0	9.0	ana



X	Y	ATT	Geometry
1.0	1.0	feli	Point(1.0, 1.0)
2.0	3.0	zcum	Point(2.0, 3.0)
7.0	7.0	ple	Point(7.0, 7.0)
8.0	3.0	sus	Point(8.0, 3.0)
8.0	9.0	ana	Point(8.0, 9.0)



Transformaciones a nivel de DAL

- Añadir una transformación a un almacén:

```
EventThemeTransform transform =  
    new EventThemeTransform();  
transform.initialize(featureStore,  
    "geometry",  
    "X",  
    "Y")  
);  
store.getTransforms().add(transform);
```

Transformaciones a nivel de DAL

- Los parámetros que necesita la transformación para aplicarse pueden ser otros almacenes de datos, como por ejemplo en Join.
- La clase **JoinTransform** implementa esta transformación.

X	Y	ATT
1.0	1.0	1
2.0	3.0	2
7.0	7.0	3
8.0	3.0	4
8.0	9.0	5

ATT	Z	T
6	1.0	4
7	3.0	5
1	7.0	7
3	3.0	6
5	9.0	9





Índice de la Presentación

1. Definición de transformación.

1.1. Transformaciones a nivel de DAL.

1.2. Transformaciones a nivel de aplicación.

2. El wizard de las transformaciones.

2.1. Uso del wizard.

2.2. Creación de un nuevo wizard.

Transformaciones a nivel de aplicación

- Para poder ser utilizadas a nivel de aplicación las transformaciones se tienen que registrar utilizando algún mecanismo de registro.

EventThemeTransform

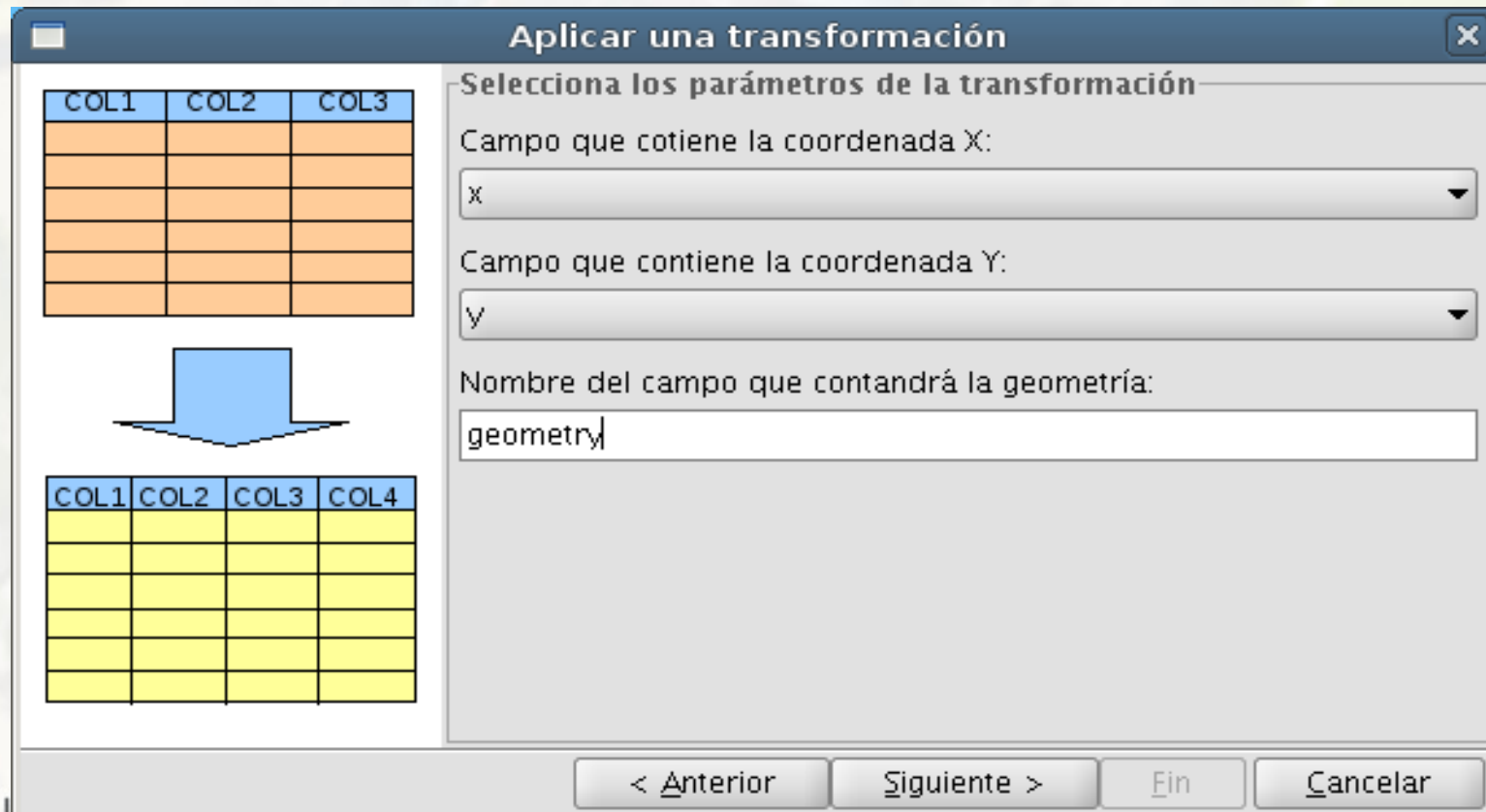
JoinTransform

Registro



Transformaciones a nivel de aplicación

- Cada transformación se tiene que inicializar con unos parámetros concretos, por lo que necesita un formulario concreto.



Aplicar una transformación

Selecciona los parámetros de la transformación

Campo que contiene la coordenada X:
x

Campo que contiene la coordenada Y:
y

Nombre del campo que contendrá la geometría:
geometry

< Anterior Siguiente > Fin Cancelar

Transformaciones a nivel de aplicación

- Se ha definido un mecanismo que permite registrar tanto las transformaciones como el interface de usuario que las configura.

EventThemeTransform
+
GUI

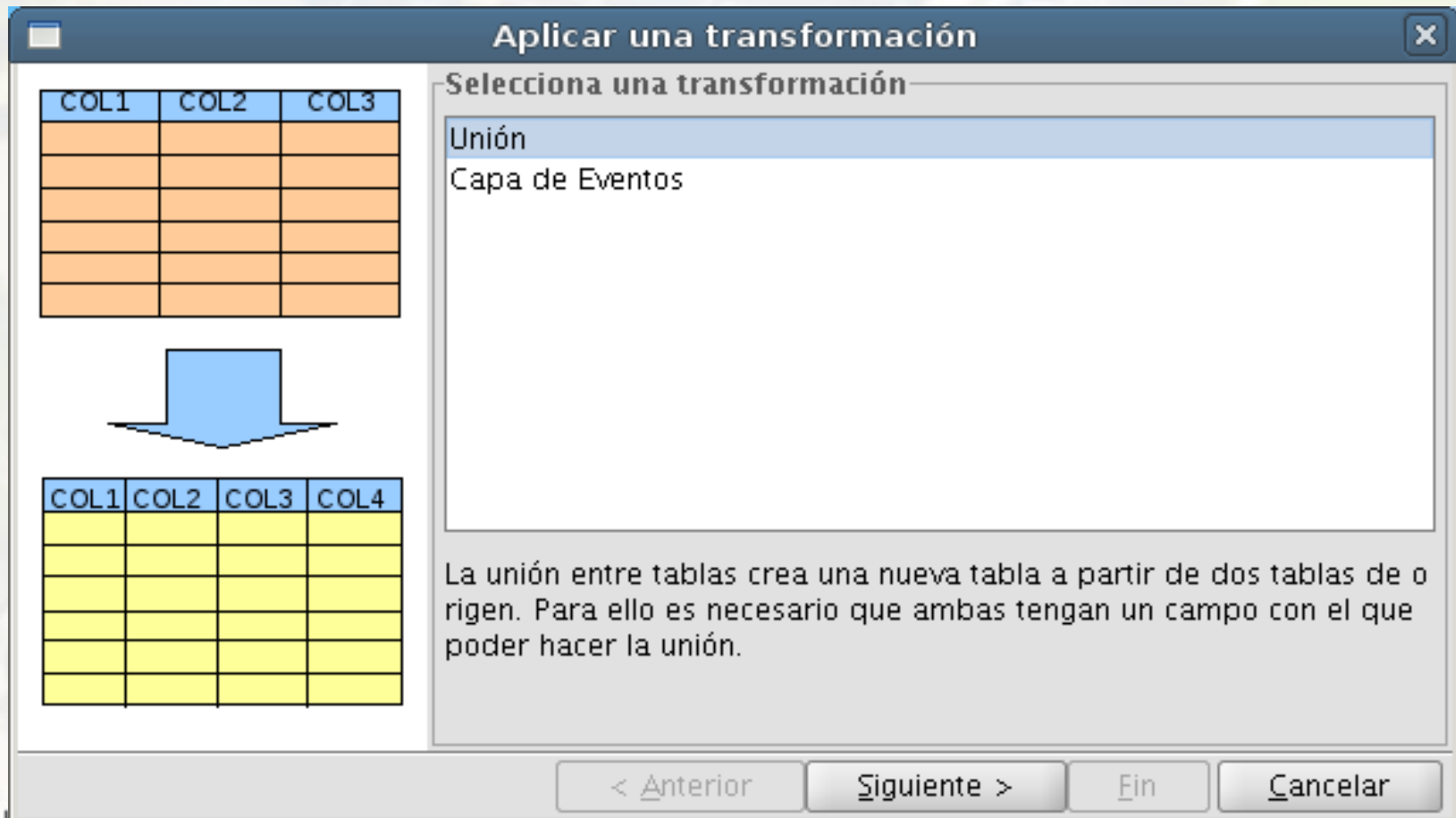
JoinTransform
+
GUI

Registro



Transformaciones a nivel de aplicación

- Para poder aplicar las transformaciones de forma genérica, se ha creado un wizard.



Aplicar una transformación

Selecciona una transformación

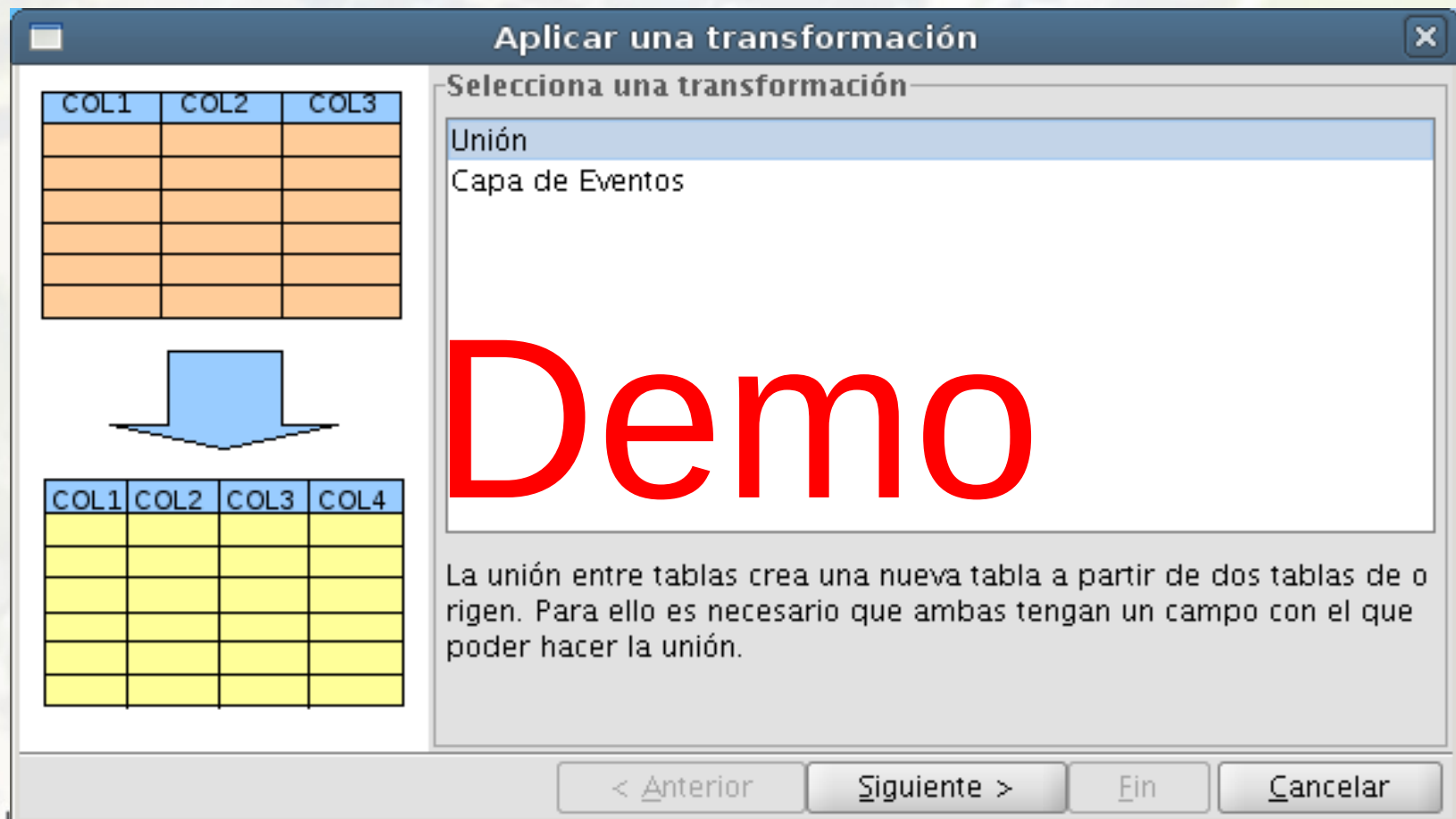
- Unión
- Capa de Eventos

La unión entre tablas crea una nueva tabla a partir de dos tablas de n ríngen. Para ello es necesario que ambas tengan un campo con el que poder hacer la unión.

< Anterior Siguiente > Fin Cancelar

Transformaciones a nivel de aplicación

- Para poder aplicar las transformaciones de forma genérica, se ha creado un wizard.



Aplicar una transformación

Selecciona una transformación

Unión
Capa de Eventos

Demo

La unión entre tablas crea una nueva tabla a partir de dos tablas de n rigen. Para ello es necesario que ambas tengan un campo con el que poder hacer la unión.

< Anterior Siguiente > Fin Cancelar

Transformaciones a nivel de aplicación

- **DataTransformLocator**: Se trata del locator de la librería. Nos proporciona los servicios de localización del DataTransformManager a usar por la librería.
- **DataTransformManager**: Se trata del punto que nos da acceso al API de transformaciones en gvSIG. A partir de él podemos realizar realizar básicamente dos operaciones:
 - Gestionar el wizard de transformaciones.
 - Gestionar las transformaciones.

Transformaciones a nivel de aplicación

- Para obtener la instancia de manager hay que ejecutar el siguiente código:

```
DataTransformManager transformManager =  
DataTransformLocator.  
getDataTransformManager();
```

Índice de la Presentación

1. Definición de transformación.

1.1. Transformaciones a nivel de DAL.

1.2. Transformaciones a nivel de aplicación.

2. El wizard de las transformaciones.

2.1. Uso del wizard.

2.2. Creación de un nuevo wizard.

El wizard de las transformaciones

- Implementa el interface DataTransformWizard.
- Se puede acceder al wizard de las transformaciones desde cualquier punto de la aplicación ejecutando:

```
DataTransformWizard transformWizard =  
transformManager.createWizard();
```



El wizard de las transformaciones

- Tiene que devolver una IWindow que gvSIG sea capaz de dibujar.

```
IWindow window =  
    transformWizard.getWindow();  
PluginServices.getMdiManager().  
    addWindow(window);
```

Índice de la Presentación

1. Definición de transformación.

1.1. Transformaciones a nivel de DAL.

1.2. Transformaciones a nivel de aplicación.

2. El wizard de las transformaciones.

2.1. Uso del wizard.

2.2. Creación de un nuevo wizard.

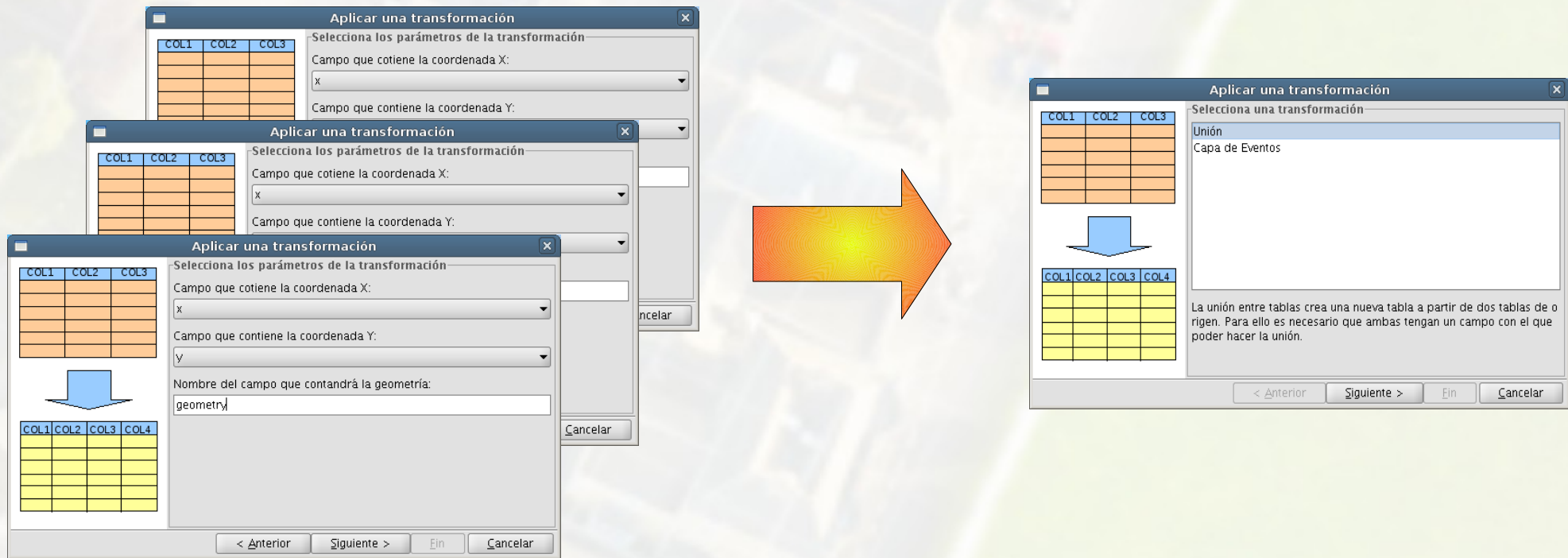
Creación de un nuevo wizard

- Se puede reemplazar el wizard por defecto de gvSIG por otro nuevo wizard.
- Para ello, hay que crear una clase que extienda de **DataTransformWizard** y registrarla:

```
transformManager.  
    registerDataTransformManager(  
        MyDataTransformManager.class);
```

Creación de un nuevo wizard

- Los paneles personalizados necesitan información del wizard.



Creación de un nuevo wizard

- Además de ofrecernos un interfaz de usuario (IWindow), el wizard deberá ser capaz de:
 - Proporcionar una transformación de entre todas las transformaciones registradas que hay que aplicar.
 - Proporcionar un DataStore se entre todos los que tiene la aplicación para ejecutar la transformación.
 - Proporcionar algunos métodos de control para poder gestionar el wizard (p.e: habilitar el botón siguiente)

Índice de la Presentación

3.Las transformaciones.

3.1. Creación de una transformación.

3.2. Los panelas asociados a la transformación.



Las transformaciones

- Al hablar de transformaciones a nivel de aplicación nos estamos refiriendo a transformación a nivel de DAL + GUI.
- La clase que implementa una transformación a nivel de aplicación es **DataTransformGui**.
- Este interfaz proveerá al wizard de transformaciones de toda la información que necesita para poder ser seleccionada y poder aplicarse.

Las transformaciones

Aplicar una transformación

Selecciona una transformación

COL1	COL2	COL3

↓

COL1	COL2	COL3	COL4

Unión

Capa de Eventos

`getName()`

`getDescription()`

La unión entre tablas crea una nueva tabla a partir de dos tablas de origen. Para ello es necesario que ambas tengan un campo con el que poder hacer la unión.

< Anterior Siguiete > Fin Cancelar

Las transformaciones

getPanels()

Aplicar una transformación

Selecciona los parámetros de la transformación

Campo que contiene la coordenada X:
x

Campo que contiene la coordenada Y:
y

Nombre del campo que contendrá la geometría:
geometry

COL1	COL2	COL3

↓

COL1	COL2	COL3	COL4

< Anterior Siguiente > Fin Cancelar

Las transformaciones

Aplicar una transformación

COL1	COL2	COL3

↓

COL1	COL2	COL3	COL4

Aplicar la transformación

Ha aplicado una transformación sobre una tabla. La transformación resultante tiene un atributo de tipo geometría, por lo que se podrá cargar en la vista activa.

Cargar el resultado de la transformación en una capa

Selecciona la vista en la que quiere cargar la transformación

Sin título - 0

`createFeatureStoreTransform`

< Anterior Siguiente > **Fin** Cancelar



Índice de la Presentación

3. Las transformaciones.

3.1. Creación de una transformación.

3.2. Los paneles asociados a la transformación.



Creación de una nueva transformación

- Primero hay que crear una transformación a nivel de DAL.
- Posteriormente hay que crear una clase que implemente el interfaz **DataTransformGui** que:
 - Devuelva el nombre y la descripción de la transformación.
 - Aporte los paneles personalizados de la transformación.
 - Cree la transformación a partir de un **FeatureStore**.



Las transformaciones

- Para registrar las transformaciones tenemos que utilizar el **DataTransformManager**.
- La transformación tiene que heredar de **DataTransformGui**.

```
transformManager.registerDataTransform(  
    MyDataTransform.class);
```













Creación de una nueva transformación

El tema de eventos

- Se ha creado una transformación particular a nivel de aplicación para el tema de eventos: **EventThemeTransformGui**
- Para ello se ha creado un panel (**EventThemeTransformPanel**) personalizado, que tiene métodos para recoger toda la información que necesita el tema de eventos.

Creación de una nueva transformación

El tema de eventos

- ▼  > extDalTransformEventTheme 29432 [<https://gvSIG.org>]
- ▶  src-test 29164
- ▼  > src 29164
 - ▼  > org.gvsig.app.eventtheme 29164
 - ▶  EventThemeTransformExtension.java 29627
 - ▼  > org.gvsig.app.eventtheme.dal.feature 28895
 - ▶  > EventThemeTransform.java 29715
 - ▼  > org.gvsig.app.eventtheme.daltransform 29164
 - ▶  > EventThemeTransformGui.java 29627
 - ▶  > EventThemeTransformPanel.java 29627

Creación de una nueva transformación

- Creación de la transformación

```
private EventThemeTransformPanel panel;  
public FeatureStoreTransform  
    createFeatureStoreTransform(  
        FeatureStore featureStore){  
EventThemeTransform transform =  
    new EventThemeTransform();  
transform.initialize(featureStore,  
                    panel.getGeometryName(),  
                    panel.getXName(),  
                    panel.getYName());  
    return transform;  
}
```

Índice de la Presentación

3. Las transformaciones.

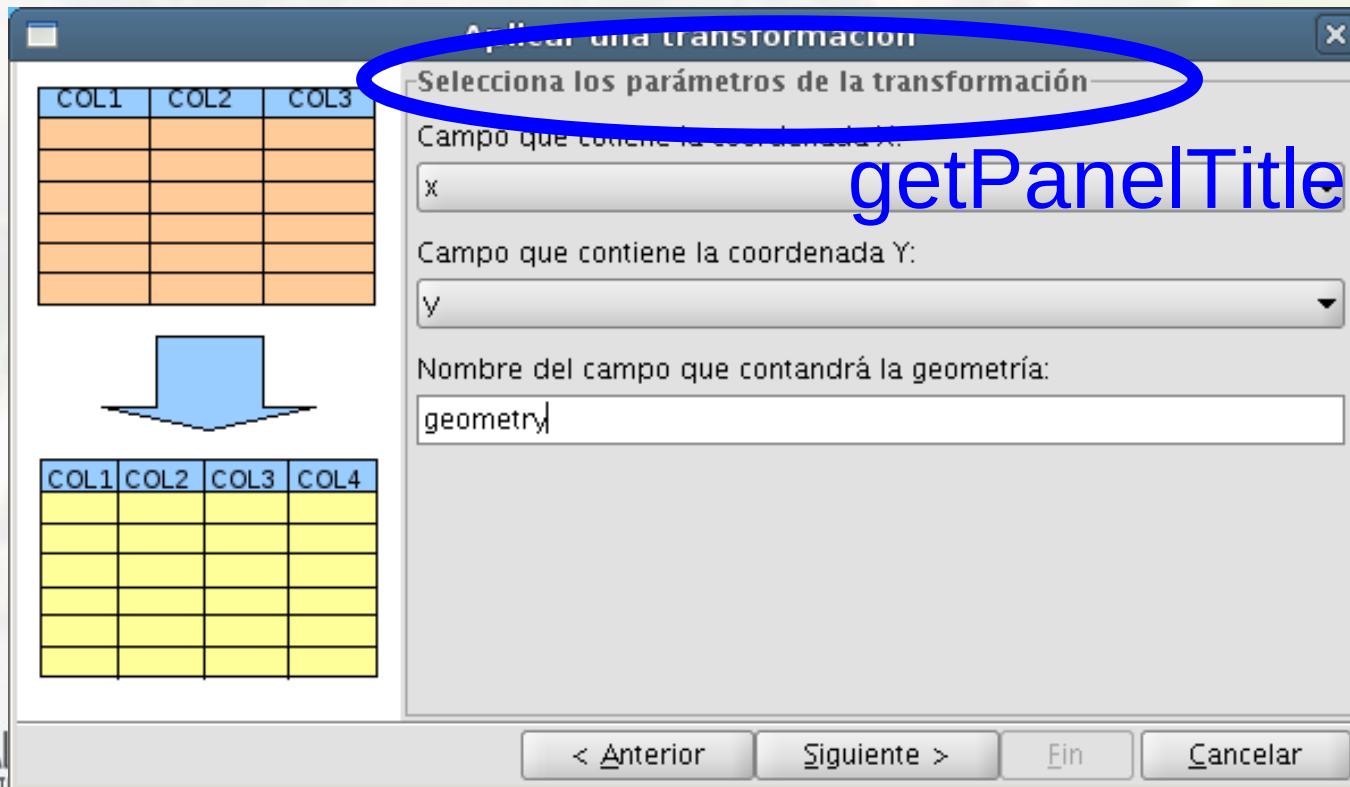
3.1. Creación de una transformación.

3.2. Los paneles asociados a la transformación.

Los paneles asociados a la transformación

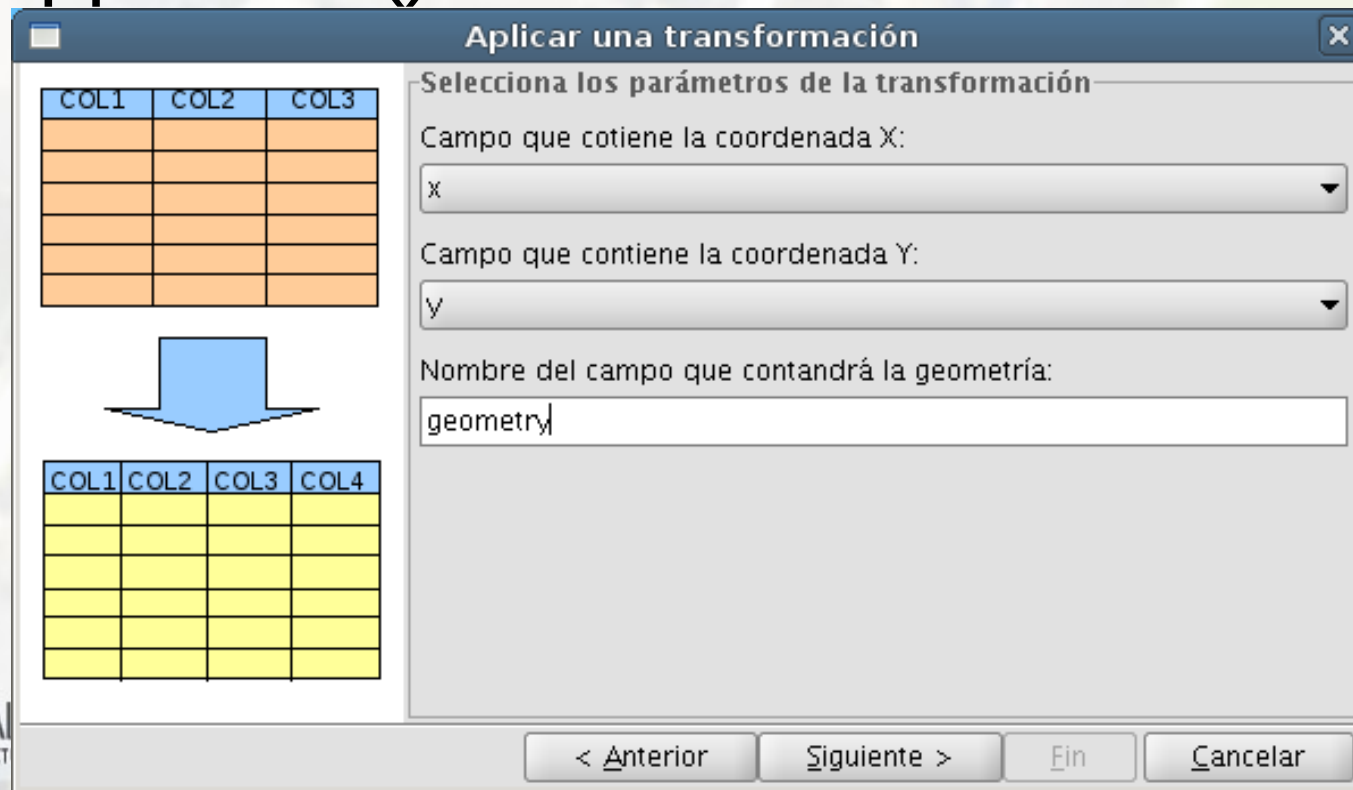
- Los paneles que aporta una transformación deben implementar el interfaz `DataTransformWizardPanel`.

`getJPanel()`



Los paneles asociados a la transformación

- Los paneles internos del el wizard, tendrán una referencia al **DataTransformWizard**.
- Podrán configurarlo mediante el método **setApplicable()**.



Los paneles asociados a la transformación

- Los paneles definen eventos para poder configurarse en función de la acción de wizard que se esté ejecutando:
 - `updatePanel()`: se ejecuta cuando el wizard llega a la posición que ocupa un panel.
 - `nextPanel()`: se ejecuta cuando el wizard avanza al siguiente panel.
 - `lastPanel()`: se ejecuta cuando el wizard retrocede al panel anterior.



Conclusiones

- Existen transformaciones a nivel de DAL que se puede crear para cambiar la forma de visualizar un origen de datos.
- Existen transformaciones a nivel de aplicación que están formadas por una transformación a nivel de DAL + GUI.

Transformación a nivel de aplicación

DAL Transform
+
GUI



Conclusiones

- Las transformaciones a nivel de aplicación se tienen que registrar en gvSIG para poder ser utilizadas.
- Las transformaciones se aplicarán sobre cualquier origen de datos que gvSIG sea capaz de manejar.

gvSIG. Geographic Information System of the Valencian Government

Copyright (C) 2007-2009 Infrastructures and Transports Department
of the Valencian Government (CIT)

This file is free documentation; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.