



# *Los fenómenos de una fuente de datos tabular*

## Índice de contenido

<b>1 Acceso a los fenómenos de una fuente de datos tabular.....</b>	<b>3</b>
<b>2 La colección de fenómenos. La clase FeatureSet.....</b>	<b>3</b>
<b>3 Manejo de la selección. La clase Selection.....</b>	<b>4</b>
<b>4 El objeto Feature.....</b>	<b>5</b>
<b>5 Modificación de los atributos de un fenómeno.....</b>	<b>5</b>
<b>6 Añadir un registro a la fuente de datos.....</b>	<b>6</b>

## 1 Acceso a los fenómenos de una fuente de datos tabular

Para acceder a los fenómenos de una fuente de datos tabular se utiliza el método *features*. Este método devuelve un objeto que nos permite iterar sobre el conjunto de datos de la capa. Sin embargo, hay que tener en cuenta que el iterador reutiliza la instancia para el nuevo valor, por lo que durante una iteración no debemos guardarnos el valor de la instancia, si no obtener una copia de esta mediante el método *getCopy* del fenómeno. El motivo para reutilizar la instancia es que esta técnica aumenta la velocidad de ejecución y crea muchas menos instancias de objetos, lo que mejora el rendimiento de la máquina virtual de java.

## 2 La colección de fenómenos. La clase FeatureSet

El código para obtener los fenómenos de la capa activa sería

```
layer = currentLayer()
features = layer.features()
```

Los métodos más importantes que tiene el objeto *featureSet* son

```
- getCount()
- update(feature)
```

- *getCount()*: devuelve el número de fenómenos que tiene la fuente de datos
- *update(feature)*: actualiza un fenómeno dentro de la colección.

Veamos cómo recorrer los fenómenos de una capa con un ejemplo. Supongamos que tenemos una capa activa cuyos fenómenos tienen una propiedad *ELEVACION*, nosotros queremos obtener el objeto *feature* que se corresponde con el valor máximo de este atributo.

Los pasos que se siguen en el ejemplo son (*ejemplo 1*):

- Obtenemos la capa activa
- Obtenemos el iterador del conjunto de fenómenos de la capa
- Inicializamos las variables
- Recorremos el conjunto de fenómenos mediante el iterador
- Comprobamos el valor del atributo *ELEVACION*
- Si es mayor que el que ya hemos almacenado guardamos el fenómeno

```
layer = currentLayer()
features = layer.features()
fmax = 0.0
newFeature = None
for feature in features:
    if feature.ELEVACION > fmax:
        newFeature = feature.getCopy()
#
#resto de codigo
```

Recuerda que debes realizar una copia de la feature y no una asignación directa.

### 3 Manejo de la selección. La clase Selection

Podemos obtener un subconjunto concreto de fenómenos de una capa o tabla aplicando un filtro sobre los atributos de los fenómenos. Para ello tenemos que invocar al método 'features' de la capa usando la condición que queremos que se aplique al conjunto de los datos para obtener nuestro subconjunto.

Por ejemplo, el código de aplicar un filtro a nuestro conjunto de datos podría ser (ejemplo 2):

```
layer = currentLayer()
expresion = "ID >= 50 AND ID < 100"
features = layer.features(expresion)
#
#resto de codigo
```

También puede darse la circunstancia de que queramos operar con un conjunto de datos que ya tenemos seleccionados en la capa. En este caso deberemos usar el método getSelection de la capa (ejemplo 3).

```
layer = currentLayer()
features = layer.getSelection()
#
#resto de codigo
```

Es posible, también, que queramos recorrer los fenómenos e ir añadiendo algunos a la selección, para ello, lo primero que debemos hacer es obtener el objeto 'selection' y añadir los fenómenos que queramos mediante el método 'select' (ejemplo 4).

```
layer = currentLayer()
features = layer.features()
for feature in features:
    featureCopy = feature.getCopy()
    layer.getSelection().select(featureCopy)
#
#resto de codigo
```

Este ejemplo anterior, haría una selección de todos los fenómenos de la capa, lo que es equivalente al método 'selectAll' (ejemplo 5).

```
layer = currentLayer()
features = layer.getSelection().selectAll()
```

Si queremos seleccionar los fenómenos que cumplan una determinada condición podemos hacerlo de la siguiente manera. Este ejemplo añade a la selección el fenómeno que tiene el valor más alto en el campo ELEVATION (ejemplo 6).

```
layer = currentLayer()
features = layer.features()
```

```
fmax =0
featureCopy = None
for feature in features:
    if feature.ELEVACION > fmax:
        featureCopy = feature.getCopy()
if featureCopy:
    layer.getSelection().select(featureCopy)
#
#resto de codigo
```

## 4 El objeto Feature

Representa un fenómeno de una fuente de datos tabular. Sus métodos principales son:

```
- geometry()
- getValues()
- edit()
- set(nombre, valor)
```

- `geometry()`: Devuelve la geometría por defecto del fenómeno.
- `getValues()`: Devuelve un diccionario con el nombre y el valor de las propiedades del fenómeno
- `edit()`: pone el fenómeno en modo edición.
- `set(nombre, valor)`: Establece en la propiedad establecida mediante el parámetro nombre el valor pasado como parámetro.
  - nombre, string: Nombre de la propiedad del fenómeno
  - valor, object: Valor que debe asignarse a la propiedad

## 5 Modificación de los atributos de un fenómeno

Los fenómenos no son editables en sí mismos. Para editar un fenómeno es necesario invocar al método `edit()`, a continuación se realizan los cambios que se quieran aplicar sobre el fenómeno.

*Internamente la edición de una feature se realiza mediante una copia de esta para evitar que se produzcan ambigüedades en el estado interno del fenómeno.*

El código para modificar un atributo del fenómeno sería (ejemplo 7)

```
fuente = currentTable()
features = fuente.features()
for feature in features:
    feature.edit()
    feature.set("NombreDelAtributo", valor)
    fuente.update(feature)

fuente.commit()
```

*El código para modificar una capa es el mismo llamando a la función `currentLayer`.*

## 6 Añadir un registro a la fuente de datos

Para añadir un registro o fenómeno nuevo a la fuente de datos tabular debe usarse el método *append* de la instancia de la fuente de datos.

```
table = currentTable()
table.append(values)
table.update()
table.commit()
```

- `append(values)`: Crea un nuevo fenómeno y lo añade a la colección de datos.
  - `values, dict`: Añade en la propiedad *key* del fenómeno, el valor correspondiente.

Si el objeto no está en estado de edición al usar este método se cambiará el estado a modo edición.

Por ejemplo, supongamos que tenemos un dxf sin registros con un campo "Id", de tipo entero y queremos añadir 10 registros nuevos cuyo id se incremente en una unidad (*ejemplo 8*).

```
table = currentTable()
values = dict()
for i in range(1,11):
    values["Id"] = i
    table.append(values)
table.commit()
```

### gvSIG Association

Plaza Don Juan de Villarrasa 14-5,  
46001, Valencia (Spain)

Registro Nacional de Asociaciones: 596206

e-mail : [info@gvsig.com](mailto:info@gvsig.com)

Web: [www.gvsig.com](http://www.gvsig.com)

Web del proyecto: <http://www.gvsig.org>

Documentación realizada por Víctor Acevedo.

#### Listas de Distribución

Existen tres listas de distribución con el objeto de facilitar la comunicación entre todos los interesados en el proyecto gvSIG. Las dos primeras, la de usuarios y la de desarrolladores, están principalmente orientadas a la comunidad de habla hispana, siendo el castellano el idioma preferente a utilizar en las mismas. La tercera de ellas, lista internacional, está orientada principalmente al resto de comunidades y la lengua preferente a utilizar es la inglesa.

- **Lista de usuarios.** Aquí podéis hacer llegar vuestra opinión sobre el funcionamiento: qué cosas os gustaría que se desarrollaran, dudas en el uso de gvSIG y todo aquello que penséis que tiene cabida en una lista de usuarios. El enlace para la suscripción a la lista de usuarios es:

[http://listserv.gva.es/mailman/listinfo/gvsig\\_usuarios](http://listserv.gva.es/mailman/listinfo/gvsig_usuarios)

- **Lista de desarrolladores.** Está orientada para todos los interesados en conocer cómo está desarrollado el gvSIG. El enlace para la suscripción a esta lista es:

[http://listserv.gva.es/mailman/listinfo/gvsig\\_desarrolladores](http://listserv.gva.es/mailman/listinfo/gvsig_desarrolladores)

- **Lista internacional.** Está orientada tanto para usuarios como para desarrolladores de habla no hispana. El idioma a utilizar es preferentemente inglés. El enlace para la suscripción a esta lista es:

[http://listserv.gva.es/mailman/listinfo/gvsig\\_internacional](http://listserv.gva.es/mailman/listinfo/gvsig_internacional)

Todos los nombres propios de programas, sistemas operativos, equipo hardware etc., que aparecen en este curso son marcas registradas de sus respectivas compañías u organizaciones.

© 2013 gvSIG Association

Este manual se distribuye con la licencia Creative Commons Reconocimiento-CompartirIgual 3.0 Unported (<http://creativecommons.org/licenses/cc-by-sa/3.0/deed.es>) – Ver condiciones en Anexos