





# Práctica 3: Configuración avanzada de proyectos CartoDruid

### 1 Contenido

1	Int	roducción	2
2	Cre	eación del proyecto	2
3	De	finición de etiquetas personalizada	4
4	Sin	nbolización basada en reglas	5
5	Personalización de consultas de entidades		
	5.1	Personalización del listado de entidades	9
	5.2	Personalización del formulario de información	9
6	Ре	rsonalización del formulario de edición	11
7	De	finitionQueries	14
8	Us	uarios y seguridad en Cartodruid	15







## 1 Introducción

En esta práctica reutilizaremos el proyecto de la práctica 1 para añadir elementos más avanzados de la configuración de CartoDruid.

El objetivo es explorar estos aspectos:

- Definición de etiquetas personalizadas
- Simbolización basada en reglas
- Formularios Alpha numéricos
- Definition queries
- Posibilidad de elegir los usuarios de un proyecto

### 2 Creación del proyecto

Vamos a empezar creando un proyecto practica 3 y reutilizaremos parte de la configuración de la práctica 1.

#### Práctica:

- 1. Creamos la carpeta de proyecto practica3 con la carpeta cartodruid.
- 2. Creamos el proyecto cartodruid con el nombre practica3 e importamos las capas de recintos, MTN y Pnoa del proyecto practica 1.
- 3. Importamos también la capa "arboles", pero tenemos un problema:

En nuestra nueva tarea nos han pedido que **registremos** también la altura **y la** especie. Utilizaremos esta información para simbolizar y dar información en las etiquetas de las entidades.

- 4. **Guardamos el proyecto y lo copiamos al PC**: copiamos todas las carpetas y nos quedamos solo con las que nos interesen como en otros proyectos.
- 5. Ahora vamos a hacer una copia de arboles.sqlite, la llamaremos arboles\_mod.sqlite y vamos a cambiar el XML de capas crtdrd.practica3.xml para empezar a utilizar el nuevo fichero
- 6. Copiamos las carpetas data y config al móvil, abrimos el proyecto y comprobamos con un *long tab* que la capa ahora apunta a la nueva BD.







#### Añadir nuevo campo a la BD SQLite

Una vez que tenemos configurado el proyecto, vamos a añadir los nuevos campos a la capa de trabajo:

- 1. Abrimos la capa arboles\_mod.sqlite con QGIS, vamos a la opción de "tabla de atributos" y activamos la edición (icono de la izquierda).
- 2. Después pulsamos sobre el botón de "Campo nuevo" y creamos un campo "altura" de tipo decimal (REAL).
- 3. Añadimos otro campo "tipo" de tipo TEXT.
- 4. Guardamos la BD y la pegamos en el dispositivo y comprobamos que ahora el campo aparece.

Ŧ	Q arboles— Objetos Totales: 1, Filtrados: 1, Seleccionados: 0							
an a	🗾 🖉 🕞 😂 📾 🗮 👘 😒 🚳 🚺 😜	🗏 🛯 🔩 🍸	🗷 🏘 P 🔢 🛍 🗏 🔍 📾					
	123 PK_UID = E 123		Campo nuevo (Ctrl+W)					
	Expresión _							
•	defaultUser	PK_UID	1					
		observaciones_	tiene una plaga de .					
		c_uuid	4b77c3cd-f675-497a-b4d8-d0de5ff86904					
		c_refrec	47087000000040516900002					
		b_insp						
		f_insp						
		d_observa						
		username	defaultUser					
		latitude	41,69760375					

#### Notas

- Cartodruid construye los formularios dinámicamente en función de los campos de la BD, se pueden crear y eliminar campos de las capas utilizando los tipos de datos de SQLite.
- Cartodruid da una representación automática a los campos en función de su tipo: TEXT, INT o REAL.
- Para el caso de campos **booleanos**, cartodruid representa los campos que tienen el prefijo "b\_" como un radio-button (ej b\_insp). De igual forma, los campos que empiezan con "f\_" tienen un tratamiento especial y son representados como **fechas**.
- QGIS no utiliza tipos SQLite estrictamente válidos, pero SQlite es robusto en cuanto a la resolución de los tipos e interpreta correctamente VARCHAR como TEXT (porque contiene CHAR)

Ver Apartado *Determination Of Column Affinity* en https://sqlite.org/datatype3.html

En resumen: si manejas SQL, es recomendable utilizar Spatialite-GUI y crear las columnas con sentencias ALTER TABLE.





## 3 Definición de etiquetas personalizada

El valor que cartodruid muestra en las etiquetas puede calcularse en función de los campos de la capa. Un caso sencillo es utilizar una concatenación de campos, por ejemplo:

pk\_uid ||'-'|| observaciones

Pero realmente lo que hace Cartodruid por debajo es ejecutar una expresión SQL, por lo que podemos crear expresiones más complejas para calcular en texto a mostrar.

#### Práctica

- 1. Ir al menú de la capa de árboles, editar la capa y establecer que la etiqueta se calcule como la concatenación de los campos observaciones y altura.
- 2. Editar un par de geometrías y activar/desactivar las etiquetas para ver el cálculo.

#### Práctica avanzada:

Utilizar la función SQLite DECODE para mostrar el texto "muy alto" si la altura es > 10 metros.

1. Esta sería la expresión del campo:

```
CASE WHEN altura >10 THEN 'muy alto' ELSE '' END
```

 Hay que editar el XML de la capa para establecer este valor en la etiqueta <labelExpression>

#### Notas

- Las etiquetas se activan y desactivan en el menú de la capa, pero aparte, cartodruid permite definir los niveles a los que se visualizarán las etiquetas para evitar la sobrecarga al renderizar cada texto de cada geometría.
- Las etiquetas se aplican tanto al mapa como al panel de información inferior, aquí vemos que los recintos tienen una etiqueta calculada para mostrar el uso SIGPAC y el coeficiente CAP.





## 4 Simbolización basada en reglas

Junta de

Castilla v León

En este caso vamos a dar estilos diferentes a los árboles en función del campo de altura.

Cartodruid permite definir tanto los estilos como las reglas para aplicarlos en función de los campos de la capa.

En el **Apartado 6. Configuración de simbologías en proyectos** del manual de referencia se describe cómo personalizar los estilos de los colores de bordes, fondos, fuentes de etiquetas, etc.



Ilustración 1 Relación entre estilos y simbologías

Cartodruid simplifica este proceso utilizando unas simbologías por defecto que muchas veces son suficientes para la mayoría de los proyectos. En la simbolización solo tenemos que elegir el nombre del estilo que queremos utilizar Ej: *Fernando, Laura, Pilar, Carmen,* etc.

Se pueden crear tanto estilos por proyecto como modificar los estilos por defecto editando los ficheros config/crtdrdSymbologies.proyecto.xml.

#### Práctica: Dos capas sobre una misma base de datos SQLite

En este apartado vamos a **copiar una capa en el XML** y **simbolizarla con reglas** en función el código SIGPAC.

#### https://www.fega.gob.es/es/content/%C2%BFqu%C3%A9-usos-se-distinguen-en-sigpac

Queremos dar el color verde a los pastos (PA, PR), tierras arables (TA) y uso Forestal (FO), y el resto un color por defecto. Usa estas simbologías: ALICIA, JAIME, CAMILLO, PABLO, SOFIA.





2023AGR018 - POSICIONAMIENTO POR SATÉLITE Y SIG EN DISPOSITIVOS MÓVILES. CARTODRUID

Pero en lugar de modificar la capa de SIGPAC que existe, vamos a añadir una capa más a la TOC y a esta capa sigpacmod la vamos a simbolizar por el código de uso SIGPAC.

- 1. Antes de editar el XML siempre hay que hacer una copia antes.
- Editar el fichero crtdrdLayers.practica3.xml y copiamos el trozo de XML de los recintos para crear una capa sigpac\_mod. Al copiarla, hay que modificar nombre e identificador en los campos señalados:



3. Después copiamos el fichero al dispositivo y se debería ver la capa nueva.



#### Nota:

 Lo que estamos haciendo en este caso es tener dos capas que apuntan a la misma base de datos, en ocasiones nos interesa tener filtrados unos elementos y superponerlos con otros de la misma capa que puedan estar relacionados, o ver la misma información con dos criterios de colores diferentes.







#### Práctica: Simbolización basada en reglas

En este apartado vamos a **simbolizarla con reglas** la nueva capa en función el código SIGPAC.

https://www.fega.gob.es/es/content/%C2%BFqu%C3%A9-usos-se-distinguen-en-sigpac

Queremos dar el color verde a los pastos (PA, PR), tierras arables (TA) y uso Forestal (FO), y el resto un color por defecto. Usa estas simbologías: ALICIA, JAIME, CAMILLO, PABLO, SOFIA.

 Tenemos que editar el fichero de capas y añadir dentro de la etiqueta de la capa <es.jcyl.ita.crtcyl.core.model.VectorialLayer>, un trozo de código XML como este:

```
<entry>
  <string>recintos 4258 20240102</string>
  <es.jcyl.ita.crtcyl.core.model.VectorialLayer>
    . . .
    <symbologies>
      <!-- Define un bloque como este para cada regla -->
      <es.jcyl.ita.crtcyl.core.model.style.ConditionalSymbology>
        <id>JAIME</id>
        <name>Uso Forestal</name>
        <condition>c_uso_sigpac = 'FO'</condition>
      </es.jcyl.ita.crtcyl.core.model.style.ConditionalSymbology>
      <es.jcyl.ita.crtcyl.core.model.style.ConditionalSymbology>
        <id>ALICIA</id>
        <name>Uso Pastos</name>
        <condition>c_uso_sigpac = 'PA' OR c_uso_sigpac = 'PR'</condition>
      </es.jcyl.ita.crtcyl.core.model.style.ConditionalSymbology>
    </symbologies>
  </es.jcyl.ita.crtcyl.core.model.VectorialLayer>
</entry>
```

- 2. Modifica los rangos de visualización de la capa SIGPAC para ver los municipios a menor nivel de zoom.
- 3. Pulsar sobre icono de simbología para ver la leyenda y simbología que has definido.

#### Notas

 Cartodruid siempre aplica un estilo por defecto a las geometrías. Puedes definir las reglas solamente para las geometrías que quieres que tenga un estilo diferente y "el resto" quedará con el estilo base.





2023AGR018 - POSICIONAMIENTO POR SATÉLITE Y SIG EN DISPOSITIVOS MÓVILES. CARTODRUID





#### Práctica avanzada: Simbolización basada en reglas 2

1. Añade dos nuevos estilos para que el viñedo con CAP\_RESULTANTE <=90 tenga un estilo y con CAP superior tenga un estilo diferente.

Ten en cuenta que los caracteres < y > hay que convertirlas a &gt; o &lt;

2. Localiza desde la lista de entidades de Cartodruid recintos de viñedo para comprobar que funciona la simbología.

#### Práctica avanzada: Simbolización basada en reglas 3

Cartodruid también permite definir una expresión para calcular la simbología, de forma similar a lo que vimos para la etiquetas, utilizando el tag <symbologyExpression>:

Por ejemplo, en este caso estamos dando una simbología diferente a las entidades ya inspeccionadas.

```
<symbologyExpression>case when B_INSP = 1 then &apos;GABRIEL&apos; else &apos;ALICIA&apos; end</symbologyExpression>
```

Define una regla en la capa sigpac\_mod para dar un estilo diferente a los árboles con > 10m de altura.

#### 2023AGR018 - POSICIONAMIENTO POR SATÉLITE Y SIG EN DISPOSITIVOS MÓVILES. CARTODRUID

# 5 Personalización de consultas de entidades

### 5.1 Personalización del listado de entidades

Por defecto cartodruid muestra en el listado de entidades todos los campos de la tabla, pero puedes modificar este comportamiento limitando el número de campos o calculando nuevos campos utilizando la etiqueta <sqlAsListView>.

Podemos limitar los campos a visualizar con una expresión como esta:

<sqlAsListView>CAMP01,CAMP02,CAMP03</sqlAsListView>

#### Práctica

1. Edita el fichero de capas y añade la etiqueta <sqlAsListView> para mostrar únicamente los campos c\_uso\_sigpac y cap\_resultante.

### 5.2 Personalización del formulario de información

De forma similar al listado, podemos modificar el formulario de información para mostrar los mismos campos, pero utilizando la etiqueta <sqlIdentify>.

<sqlIdentify>CAMP01,CAMP02</sqlIdentify>

#### Notas

- Cartodruid permite añadir en esta etiqueta expresiones SQL más complejas, igual que hemos visto para el caso de etiquetas y simbologías.
- Hay que tener en cuenta que el texto debe ser un XML bien formato, cualquier símbolo <> debe ser escapado como html entities, y las comillas también deben indicarase como "
- En la expresión podemos añadir subconsultas SQL para calcular campos cruzando con otras tablas o incluir consultas geográficas con las funciones de spatialite. Por ejemplo, en este caso mostramos las coordenadas del centroide de la parcela.











2023AGR018 - POSICIONAMIENTO POR SATÉLITE Y SIG EN DISPOSITIVOS MÓVILES. CARTODRUID



<sqlIdentify>c\_uso\_sigpac,cap\_resultante, st\_y(st\_centroid(geometry)) as &quote;Latitud&quote;, st\_x(st\_centroid(geometry)) as &quote;Longitud&quote; </sqlIdentify>



Un ejemplo de otra consulta más compleja:

<sglAsListView>C ZONA VIT &quot;Zona&quot;, C PARVIT||&quot;/&quot;||C SUBPARVIT "Pa, Notas Cartodruid siempre aplica un estilo por defecto a las geometrías. Puedes definir las reglas solamente para las geometrías que quieres que tenga un estilo diferentes y "el resto" quedará con el estilo base. rcela/Subpar", C COD TITULAR "Viticultor",CAST(SUBSTR(C REFREC,1,2) AS INTEGER)||"/"||CAST(SUBSTR(C\_REFREC,3,3) AS INTEGER)||"/"||CAST(SUBSTR(C\_REFREC,6,3) AS INTEGER)||"/"||CAST(SUBSTR(C REFREC,9,2) AS INTEGER)||"/"||CAST(SUBSTR(C\_REFREC,11,3) AS INTEGER)||"/"||CAST(SUBSTR(C REFREC,14,5) AS INTEGER)||"/"||CAST(SUBSTR(C\_REFREC,19,5) AS INTEGER) "Recinto SIGPAC", CASE B INSP WHEN 0 THEN 'No' WHEN 1 THEN 'S1' ELSE 'No' END "Inspeccionado",datetime(F\_INSP,'unixepoch', 'localtime') "Fecha de Inspección", T ACTA "Acta", T NUMERO ENTRADA "Número de Entrada",D\_OBSERVA "Observaciones", CASE B\_ALERTA\_SAN WHEN 0 THEN 'No' WHEN 1 THEN 'Sí' ELSE 'No' END "Alerta Sanitaria" , CASE B ALERTA REND WHEN 0 THEN 'NO' WHEN 1 THEN 'Sí' ELSE 'NO' END "Alerta de Rendimiento&guot;</sqlAsListView>



### 6 Personalización del formulario de edición

Cartodruid permite personalizar el formulario de recogida de información:

- Añadir campos calculados
- Definir listas de valores

Junta de

Castilla y León

- Organizar los campos en un orden determinado y agruparlos en tabs.
- Definir reglas de validación sobre los campos y restricciones de obligatoriedad

Esta configuración se establece en el fichero values/<id\_capa>\_AlphaEdit.xml. La lista de opciones es muy amplia y está recogida en el apartado 7. Configuración de la visualización de formularios del manual de referencia.

En este curso nos vamos a centrar en dos cosas:

- Cómo organizar los campos para facilitar la recogida de información.
- Cómo configurar listas de valores para un campo.

#### Práctica: Organización de campos del formulario

Vamos a organizar los campos en dos pestañas una llamada "Registro" con la información de altura, tipo y observaciones y otra "Inspección" con los campos de b\_inspc, f\_insp y el usuario inspeccionado.

- 1. Primero localizamos en la BD los campos que queremos mostrar ya que hay que referenciarlos en el XML.
- 2. Para activar el formulario, necesitamos configurar dos cosas:
  - a. En la configuración de la capa, hay que añadir esta etiqueta:

<alphaEditFinisher>userFormAlphaEditFinisher</alphaEditFinisher>



b. Hay que añadir un fichero en la carpeta values con la especificación del formular.

Partimos del fichero plantilla idCapa\_AlphaEdit.xml, lo pegamos en la carpeta de proyecto en practica3/cartodruid/values/arboles\_AlphaEdit.xml







- 3. Copiamos ahora los dos ficherso config/crtdLayers.practica3.xml y el values/arboles\_AlphaEdit.xml al proyecto y lo abrimos, deberían verse dos pestañas vacías.
- 4. Ahora vamos a añadir en cada pestaña los campos que queremos mostrar
- 5. En la pestaña 1 añadimos los campos altura, especie y observaciones.
- 6. En la pestaña 2 añadirmos b\_insp, f\_insp, username

Hay que editar el XML y añadir dentro de cada <tab> una sección <fields> y dentro de ésta, un bloque <field> por cada campo a añadir.

⊡ <form></form>	<field></field>
<id>Form</id>	
<name>XXX</name>	<la>laentificadorunico</la>
<pre><tabs class="linked-list"></tabs></pre>	<pre><name>Texto a mostrar</name></pre>
<pre><tab></tab></pre>	
<id>tabInspec</id>	<type>tipo de campo</type>
<name>Inspección</name>	(nancistad Field) company do la PD
<pre><fields class="linked-list"></fields></pre>	<persisteurieiuscamop bu<="" de="" la="" th=""></persisteurieiuscamop>
<10/USOSIGPACQUEC/10/	
<pre><tune>TEXTS/tune&gt;</tune></pre>	l
<pre><nersistedfield>NSO_SIGPAC</nersistedfield></pre>	, ·····
-	
<pre>field&gt;</pre>	
<id>finspeccion</id>	
<name>+ Fecha de Inspección</name>	
<type>DATE </type>	
<pre><persistedfield>F_INSPECCION</persistedfield></pre>	
<editable>true</editable>	
<deletable>true</deletable>	
-	
<id>Inspección</id>	
<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	
<pre><fields class="linked-list"></fields></pre>	
-	
<pre>ejemplo&gt;<!--</pre--></pre>	
-	
L	

#### Práctica: Campo con desplegable

En el punto anterior hemos añadido campos, pero todos con formato de texto, CartoDruid permite definir qué tipo de control Android mostrar para el campo (por ejemplo permitir solo números), o establecer reglas de validación.

En este caso vamos a cambiar el <field> del campo *especie*, para que sea un desplegable y poder elegir el valor de la lista.

- 1. Para cambiar el tipo de control hay que modificar el tag <type> para establecer el valor DROPDOWN.
- 2. Para configurar los posibles valores del desplegable, cartodruid tiene tres métodos:







- choices
   Permite definir una lista de <ítems> de un desplegable asociado al campo. Por cada ítem se puede indicar el código y descripción del elemento con las etiquetas key y value. (ver ejemplo más adelante).
- choicesByFile Identifica el fichero que contiene los valores del desplegable que se asocia al campo. El fichero tiene que estar almacenado en la ruta /cartodroid/values con el nombre defichero <fieldId>.properties.
- choicesByQuery
   Consulta SQL que se utilizará para cargar los valores del desplegable. Se debe indicar el fichero de BD que se va a utilizar con la etiqueta <dbFile> (ruta relativa al directorio data, o ruta absoluta), y la consulta con la etiqueta <query>.

Vamos a elegir el segundo método, un ejemplo de esta configuración:

<field></field>	Fichero valu
<id>usoSIGPACact</id>	
<name>Uso SIGPAC Actual</name>	
<type>COMBO</type>	TA=TIERRAS
<pre><persistedfield>C_USO_SIGPAC</persistedfield></pre>	TH=HUERTA
	PS-PASTIZA

Fichero values/usoSIGPAC.properties

TA=TIERRAS ARABLES TH=HUERTA PS=PASTIZAL PR=PASTO ARBUSTIVO

- Definimos un fichero values/especie.properties con estos valores: ALA=Álamo CHO=Chopo ROB=Roble
- 4. Copiamos el fichero, reiniciamos el proyecto y probamos la edición.



## 7 **DefinitionQueries**

Junta de

Castilla v León

Al igual que otros editores GIS, cartodruid permite definir filtros aplicables a las capas. Esto ya vimos que se puede seleccionar desde pantalla directamente por el usuario, pero también es posible establecerlo en la configuración de una capa para que aparezca pre-friltrada:

Añadiendo esta etiqueta a la configuración XML de la capa mostramos solo las entidades no inspeccionadas:

<definitionQuery>b\_insp is null or b\_insp = 0</definitionQuery>

#### Configurar filtros reutilizables para una capa

Las consultas definidas en pantalla que vimos en la práctica 1, se almacenan en la caché de la aplicación (solo en le dispositivo actual), pero es posible definir un fichero con las consultas más habituales para compartirlas entre todos los técnicos.

Para ello, definimos el fichero values/<ID\_CAPA>DefinitionQueries.properties.

El fichero contendrá el nombre que aparecerá en el desplegable de filtros y la definition query que se establecerá. Por ejemplo si establecemos el fichero /cartodroid/values/ plantacionesDefinitionQueries.properties el siguiente contenido:

```
No inspeccionados=b_insp is null or b_insp = 0
Tareas zona1=c_zona = 1
Tareas zona2=c_zona = 2
Tareas zona3=c_zona = 3
```



# 8 Usuarios y seguridad en Cartodruid

Cartodruid permite configurar una pantalla de acceso para solicitar al usuario credenciales que lo identifiquen. Ahora mismo se pueden utilizar dos tipos de credenciales:

- Cuenta de Google del dispositivo
- User/pass almacenado en una BD del proyecto.

El código del usuario registrado se almacena en el campo user para saber qué usuario modificó las entidades y se utiliza también durante el proceso de sincronización para limitar el acceso a los workspaces y para establecer políticas de copia de dato: se limitan los datos que un usuario puede modificar o descargar desde la base de datos central.

#### https://www.cartodruid.es/-/configuracion-usuarios-y-seguridad

#### Notas

• El almacenamiento del password se hace con un sha-256, es suficiente para no obtener la clave, pero no para evitar un acceso no autorizado ya que se puede modificar la BD a mano, pero en muchos casos es suficiente si los datos no son críticos.