
Bauble Documentation

Versión 1.0.50

Brett Adams

marzo 22, 2018

Índice general

1. cositas buenas que se encuentran en Bauble.	3
2. Como instalar Bauble	7
3. Como utilizar Bauble	15
4. Administración	31
5. Desarrollo de Bauble	33
6. Apoyar el proyecto Bauble	39

Bauble es un programa para manejar una base de datos que represente una colección botánica: inserción, búsquedas, informes, et cetera.

Bauble es [codigo abierto y libre](#) y es distribuido bajo la [GPL, Licencia Pública GNU](#)

cositas buenas que se encuentran en Bauble.

1.1 información taxonómica

When you first start Bauble, and connect to a database, Bauble will initialize the database not only with all tables it needs to run, but it will also populate the taxon tables for ranks family and genus, using the data from the “RBG Kew’s Family and Genera list from Vascular Plant Families and Genera compiled by R. K. Brummitt and published by the Royal Botanic Gardens, Kew in 1992”. In 2015 we have reviewed the data regarding the Orchidaceae, using “Tropicos, botanical information system at the Missouri Botanical Garden - www.tropicos.org” as a source.

1.2 importar datos

Bauble le permite importar datos que se encuentren en un formato json intermedio, pensado para intercambio entre versiones de Bauble. Los datos importados se combinan a completar lo que ya se encuentra en la base de datos. Si cree, puede pedir ayuda a un profesional de Bauble para poner en el formato json intermedio de intercambio toda la información que esté en su base de datos actual, para importarla en Bauble.

1.3 sinónimos

Bauble le permite definir sinónimos para especies, géneros, familias. También esta información puede ser representada en el formato json de intercambio y puede ser importada in una base de datos Bauble.

1.4 científicamente responsable

Bauble implementa el concepto de “accesión”, conexión abstracta entre la planta física (o un grupo de plantas) y la especie o taxón a que pertenecen. Cada accesión puede asociar todo el mismo grupo de plantas que le pertenece a uno o más taxones, en el caso no haya acuerdo entre los taxónomos a disposición del jardín: cada taxónomo puede insertar su opinión en la base de datos, sin necesitar borrar la información anteriormente presente. Todas las verificaciones se encuentran en la base de datos, con fecha y firma.

1.5 ayuda a la identificación

Bauble le permite asociar fotos a plantas físicas, esto puede ayudarle a reconocer la planta en caso de que se pierda la nota, o ayudar a la identificación taxonómica si no está disponible un taxónomo.

1.6 exportar e informar

Bauble le permite exportar los datos en cualquier formato textual le parezca útil. Bauble utiliza un motor de plantillas muy flexible, que se llama “mako”, con el cual solo su imaginación es el límite a los formatos en que exportar los datos. Una vez instalado, hay varios ejemplos en la carpeta *mako*.

1.7 anotar la información

A practicamente todos objetos en la base de datos es posible asociar notas. Las notas se pueden categorizar y por supuesto utilizar en búsquedas.

1.8 jardín o herbario

manejo de ubicaciones de plantas.

1.9 historia en la base de datos

All changes in the database is stored in the database, as history log. All changes are “signed” and time-stamped. Bauble makes it easy to retrieve the list of all changes in the last working day or week, or in any specific period in the past.

1.10 búsquedas sencillas y expresivas

Bauble le permite consultar la base de datos empleando palabras clave sencillas, p.e.: el nombre del lugar o un nombre de género, o puede escribir consultas más complejas, que no llegarán a la complejidad de SQL pero que le permitirán localizar sus datos con un nivel de detalle aceptable.

1.11 independiente del sistema de gestión de base de datos

Bauble no es un sistema de gestión de base de datos, ni intenta serlo. Bauble guarda la colección en una base de datos SQL, y se puede conectar a virtualmente cualquier sistema SQL para que exista un conector SQLAlchemy. En práctica esto incluye todos los más modernos sistemas relacionales, como MySQL, PostgreSQL, Oracle. También puede trabajar con SQLite, que, para usuarios independientes es un sistema muy eficiente y por supuesto más que suficiente. En combinación con un sistema de gestión SQL, es posible construir un sistema LAMP (Linux-Apache-MySQL-Php) o similar, e incluir la colección actualizada en el sitio web de su institución.

1.12 internacionalista

El programa Bauble tuvo origen en Inglés y toda su documentación nace en ese idioma. Sin embargo, tanto el programa cuanto la documentación utilizan *gettext*, la tecnología más avanzada y difundida para la traducción del software y de la documentación. Al momento de escribir, la traducción del programa al castellano está al 86 %, la documentación algo menos que el 20 %.

1.13 independiente de la plataforma

Instalar Bauble en Windows es un proceso linear y sencillo, no tomará más tiempo que unos 10 minutos. Bauble nació en Linux y su instalación en ubuntu, fedora o debian es igualmente muy sencilla. Recién ha sido posible instalarlo en un Mac con OSX10.9.

1.14 fácil puesta al día

El proceso de instalación produce una instalación que, si deseado, se puede poner al día en menos que un minuto. Dependiendo de la cantidad de respuesta recibida, quien trabaja en Bauble produce una nueva distribución cada par de días o de vez en cuando.

1.15 pruebas unitarias

Bauble es continuamente y extensamente controlado con pruebas unitarias, algo que hace casi imposible introducir errores en el código. Cada puesta al día es automáticamente verificada por el servicio Travis Continuous Integration. La buena integración de las plataformas Travis-ci y github hace muy complicado distribuir en github algo para que travis-ci haya evidenciado un error.

Virtualmente cada cambio y extensión que se hace en Bauble, va acompañada por unas nuevas pruebas unitarias, que define el comportamiento y evidenciará en el futuro cada cambio no deseado.

1.16 adaptable/extensible

Bauble puede ser expandido a través de complementos (plugin), y puede ser adaptado según lo que necesite la institución que lo adopte.

CAPÍTULO 2

Como instalar Bauble

2.1 Instalación

bauble.classic es un programa independiente de la plataforma software pues puede funcionar en sistemas unix como Linux y MacOSX así como en Windows.

To install Bauble first requires that you install its dependencies that cannot be installed automatically. These include virtualenvwrapper, PyGTK and pip. Python and GTK+, you probably already have. As long as you have these packages installed then Bauble should be able to install the rest of its dependencies by itself.

Nota: Si sigue los siguientes pasos, se encontrará con Bauble instalado en un entorno virtual, todas la dependencias Python estarán instalada localmente y no entrarán en conflicto con otros programas Python que puedan estar el la misma computadora.

Si por cualquier razón quisiera usted borrar Bauble, lo único que tiene que hacer es borrar el entorno virtual, que es una carpeta, con todo su contenido.

2.1.1 Instalación en Linux

1. Bajar el archivo *devinstall.sh* y ejecutarlo:

```
https://raw.githubusercontent.com/Bauble/bauble.classic/master/  
↪scripts/devinstall.sh
```

Nota: El archivo de instalación no bajará ningún conector de base de datos. Si quiere utilizar otra base de datos diferente que SQLite, tendrá que instalar un conector extra.

Si tiene interés, puede examinar el archivo para ver qué es lo que hace. Muy sintético: primero examina eventuales problemas que no se pueden solucionar en un entorno virtual, luego crea el entorno con nombre *bacl*, baja todo el código fuente y conecta la copia local con la línea de producción *bauble-1.0* en github, luego instala bauble, solucionando todas dependencias Python, y finalmente crea un archivo de arranque en la carpeta *~/bin*.

Si el proceso terminó sin errores, ya puede arrancar bauble:

```
~/bin/bauble
```

Si usted desea ponerse al día con la última versión publicada:

```
~/bin/bauble -u
```

El mismo archivo *devinstall.sh* puede ser utilizado para pasar a otra línea de producción. Sin embargo al momento solo hay *bauble-1.0*.

2. Utilizando Unity, abra un terminal, arranque bauble, su icono aparecerá en el lanzador, y puede ahora pegarlo para que quede a disposición.
3. Si quiere utilizar el conector por defecto de la base de datos [SQLite](#), o si ni idea tiene de que quiere decir eso, puede ignorar este paso. Pero si ha decidido conectarse a otra base de datos, será necesario instalar el conector correspondiente.

Para utilizar [PostgreSQL](#), activar el entorno virtual e instalar *psycpg2* con las instrucciones siguientes.

```
source ~/.virtualenvs/bacl/bin/activate  
pip install -U psycpg2
```

Podría ser necesario resolver dependencias. Cómo hacerlo, depende de cual distribución Linux utiliza, y en que versión. Averigüe la documentación.

Siguiente...

Conectarse a una base de datos.

2.1.2 Instalar en MacOSX

Siendo MacOSX una versión de unix, todo funciona como en Linux, pues más o menos.

Una dificultad presente con MacOSX es que hay muchas versiones del sistema operativo, y que la puesta al día no es automática ni gratuita, así que varios sistemas siguen funcionando con versiones obsoletas.

Last time we tested, some of the dependencies could not be installed on MacOSX 10.5 and we assume similar problems would present themselves on older OSX versions. Bauble has been successfully tested with 10.7 and 10.9.

Primer paso, construir en MacOSX un entorno reconocible como unix:

1. herramienta de desarrollo: xcode. por favor averigue en la wikipedia cual es la versión de xcode adapta a su mac.
2. gestor de paquetes: homebrew (o si utiliza un sistema más viejo, tigerbrew).

siendo instalado lo de arriba, ejecutar:

```
brew doctor
```

asegúrese de haber entendido los problemas que serán reportados, y corríjalos. pygtk necesita xquartz y brew no puede solucionar la dependencia en manera automatica. use bre para instalar xquartz:

```
brew install Caskroom/cask/xquartz
```

y finalmente instalar las últimas dependencias:

```
brew install git
brew install pygtk # takes time and installs all dependencies
```

siga las instrucciones relativas a como activar lo que acaba de instalar.

ahora puede seguir como en una normal máquina unix, y para eso tenemos un archivo de instalación. Lea las instrucciones para Linux, las siga, y disfrute.

Siguiente...

Conectarse a una base de datos.

2.1.3 Instalar en Windows

The Windows installer used to be a «batteries-included» installer, installing everything needed to run Bauble. The current maintainer of bauble.classic cannot run Windows applications. If you want to run the latest version of bauble on Windows: download and install the dependencies and then install Bauble from the source package.

Por favor reporte cualquier problema. Ayuda para preparar un paquete de instalación será muy apreciada.

Nota: Bauble ha sido comprobado con W-XP, W7 y W8. Muy probablemente funciona también con otras versiones de Windows.

Nota: Direct download links are given for all needed components. They have been tested in September 2015, but things change with time. If any of the direct download links stops working, please ring the bell, so we can update the information here.

los pasos de instalación en Windows:

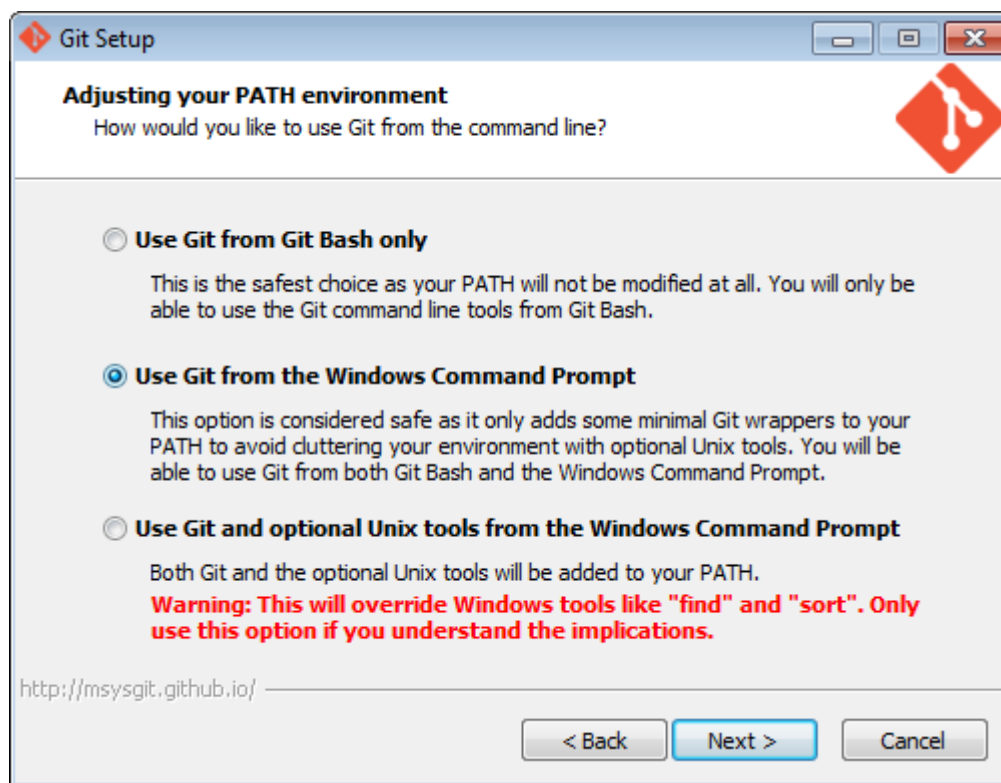
2.1. Instalación

1. baje e instale git desde:

<https://git-scm.com/download/win>

[Direct link to download git](#)

todas opciones por defecto están bien, excepto que queremos poder encontrar git desde la línea de mando Windows:



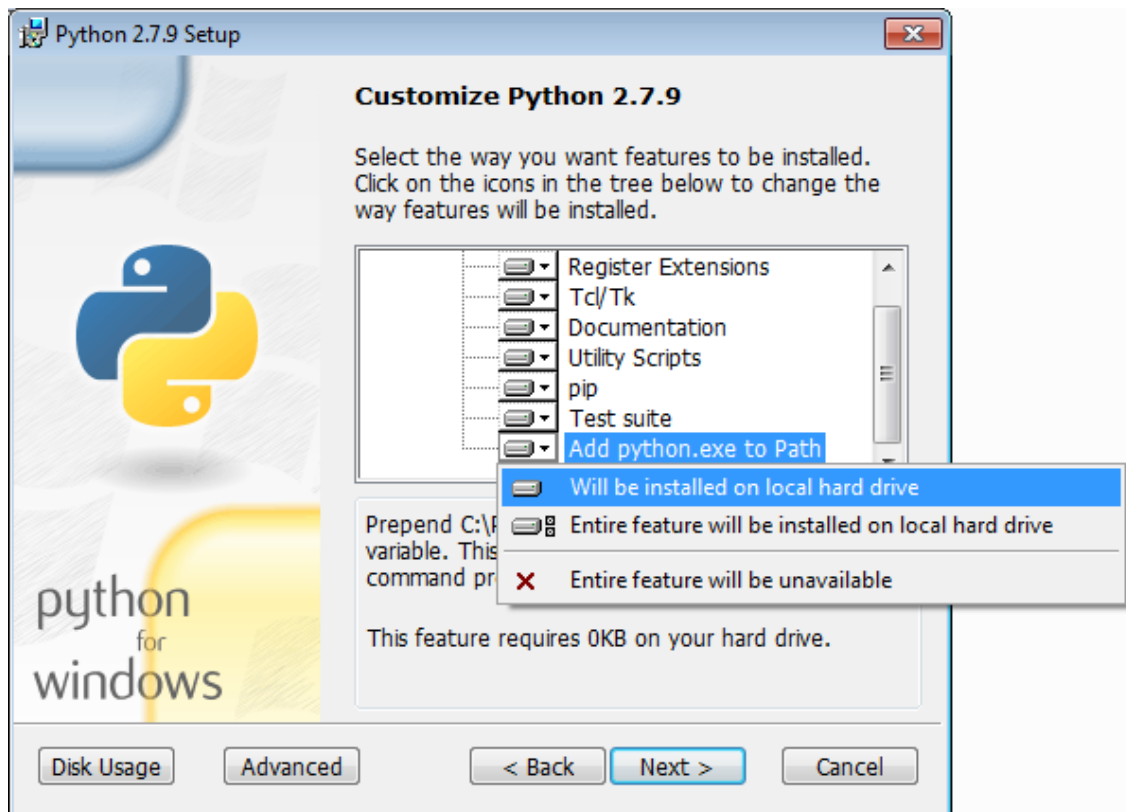
2. baje e instale Python 2.7 (32 bit) desde:

<http://www.python.org>

[Direct link to download Python](#)

Bauble has been developed and tested using Python 2.x. It will definitely **not** run on Python 3.x. If you are interested in helping port to Python 3.x, please contact the Bauble maintainers.

Cuando instala Python, ponga sus programas en el PATH:

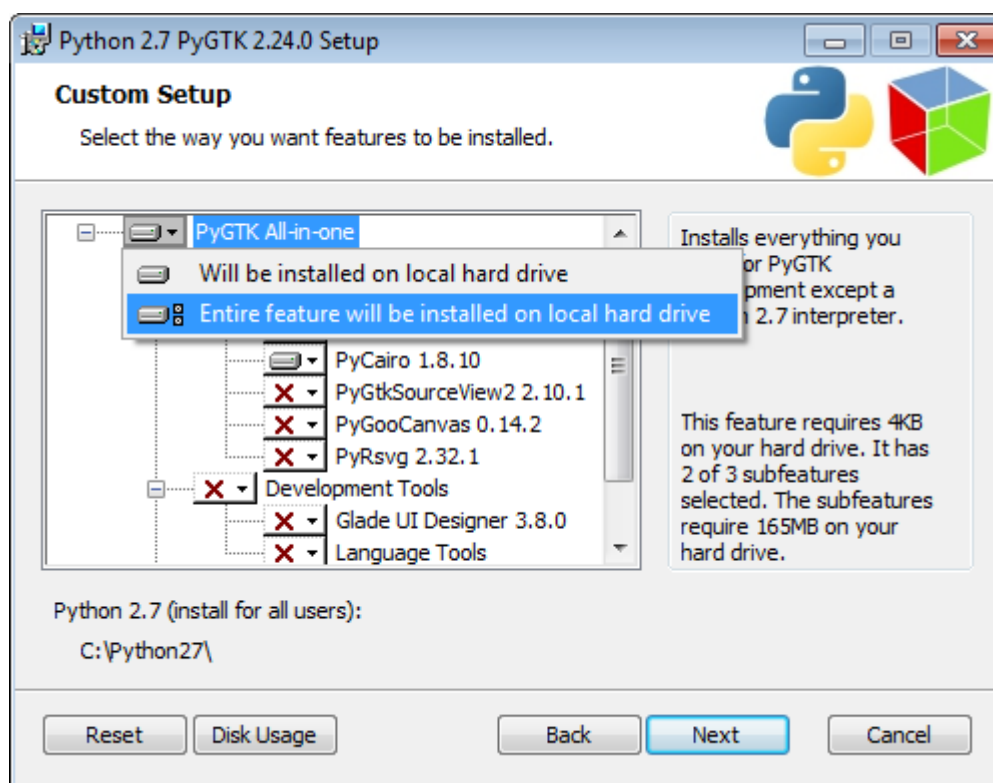


3. baje e instale pygtk desde el repositorio siguiente (se requiere python 32bit). vamos a necesitar versión «all in one»:

<http://ftp.gnome.org/pub/GNOME/binaries/win32/pygtk/>

Direct link to download PyGTK

hágase una instalación completa, incluyéndolo absolutamente todo:



4. (opcional) descargar e instalar un conector de base de datos además que `sqlite3`.

En Windows no es nada fácil instalar `psycpg2` desde las fuentes. Por favor evítese complicaciones y utilice un paquete precompilado:

<http://initd.org/psycpg/docs/install.html>

[Direct link to download psycpg2](#)

5. REINICIALICE

así es Windows, cambió cosas en el sistema, hay que reiniciar!

6. baje y ejecute el archivo batch:

```
https://raw.githubusercontent.com/Bauble/bauble.classic/master/
→scripts/devinstall.bat
```

esto pone una copia del repositorio `bauble.classic` en su equipo, en la carpeta `Local\github\Bauble`. conectada con la línea de producción 1.0, en seguida crea un entorno virtual en que finalmente instala `bauble`.

you can also run `devinstall.bat` passing it as argument the numerical part of the production line you want to follow.

7. descarga también el archivo para poner al día su instalación:

```
https://raw.githubusercontent.com/Bauble/bauble.classic/master/
→scripts/bauble-update.bat
```

las versiones de Bauble después de la 1.0.40 averiguan, al arrancar, si hay una versión más recién en el repositorio central.

cuando quiera poner al día su instalación, abra una ventana de terminal y ejecute `bauble-update.bat`

8. you can now start bauble using the `bauble.lnk` shortcut that the installation procedure copies to the `Scripts` directory of the virtual environment:

`%HOMEDRIVE%%HOMEPATH%\virtualenv\bacl\Scripts\bauble.lnk`

If you would like to generate and print PDF reports using Bauble's default report generator then you will need to download and install [Apache FOP](#). After extracting the FOP archive you will need to include the directory you extracted to in your `PATH`.

Siguiente...

Conectarse a una base de datos.

2.1.4 Troubleshooting the Install

1. What are the packages that are installed by Bauble:

The following packages are required by Bauble

- SQLAlchemy
- lxml

The following packages are optional:

- Mako - required by the template based report generator
- gdata - required by the Picasa photos InfoBox

2. Couldn't install lxml.

The lxml packages have to be compile with a C compiler. If you don't have a Make sure the libxml and libxsl packages are installed. Installing the Cython packages. On Linux you will have to install the gcc package. On Windows there should be a precompiled version available at <http://pypi.python.org/pypi/lxml/2.1.1>

3. Couldn't install gdata.

For some reason the Google's gdata package lists itself in the Python Package Index but doesn't work properly with the `easy_install` command. You can download the latest gdata package from:

<http://code.google.com/p/gdata-python-client/downloads/list>

Unzip it and run `python setup.py installw` in the folder you unzip it to.

Siguiente...

Conectarse a una base de datos.

Como utilizar Bauble

3.1 La Primera Vez

3.1.1 Que tal SQLite?

Tres preguntas: ¿esta es la primera vez que utiliza Bauble?, ¿va a trabajar desde un solo ordenador?, ¿no tiene ni meñique idea de que es un sistema de gestión de base de datos? Pues si la respuesta a una de estas es «así es», mejor se quede con SQLite, la base de datos SQL más utilizada en el mundo, fácil, rápida, sin administración.

Con SQLite, no hace falta preparar nada y puede seguir con *connecting*.

On the other hand, if you want to connect more than one bauble workstation to the same database, or if you want to make your data available for other clients, as could be a web server in a LAMP setting, you should consider keeping your database in a database management system like *PostgreSQL* or *MySQL/MariaDB*, both supported by Bauble.

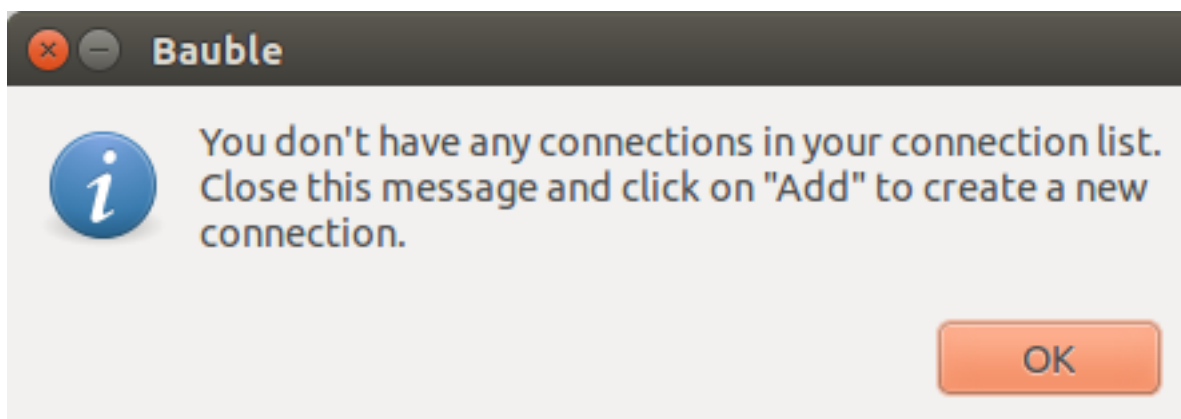
When connecting to a database server as one of the above, you have to manually create: at least one bauble user, the database you want bauble to use, and to give at least one bauble user full permissions on its database. When this is done, Bauble will be able to proceed, creating the tables and importing the default data set. The process is database-dependent and it falls beyond the scope of this manual.

If you already got the chills or sick at your stomach, no need to worry, just stick with SQLite, you do not miss on features nor performance.

3.1.2 Conectarse a una base de datos

Al arrancar Bauble, lo primero que aparece es una ventana para escoger la conexión.

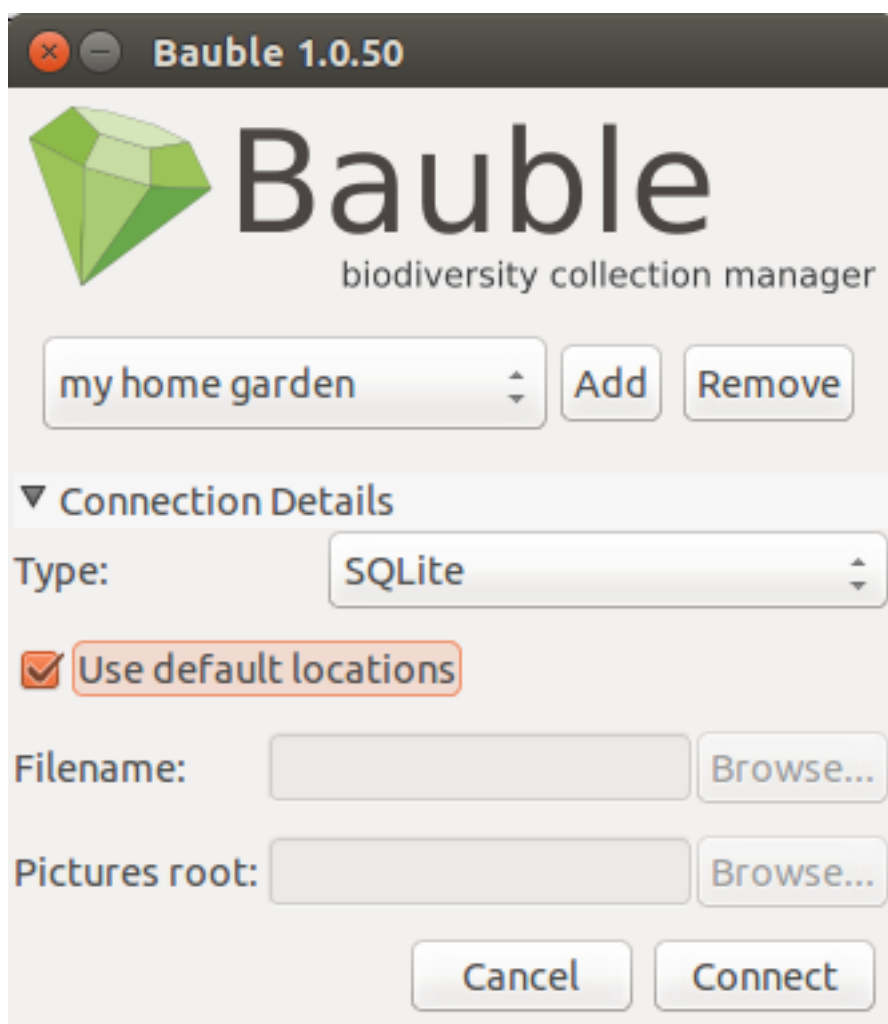
Pues si esta es la primera vez que arranca Bauble por supuesto no hay conexiones. Bauble se lo deja notar.



Como Bauble le sugiere: cierre la ventana, con esto vuelve a la ventana de conexión, donde puede pisar **Add** para crear la primera conexión.



No más inserte el nombre de la conexión, algo sencillo y relevante, que se asocie a la colección que va a poner en la base de datos (por ejemplo: «parques municipales»), y pise **OK**. Bauble vuelve a la ventana anterior, donde el nombre de la conexión está seleccionada y los Detalles de conexión se han expandido.



especificar los detalles de conexión

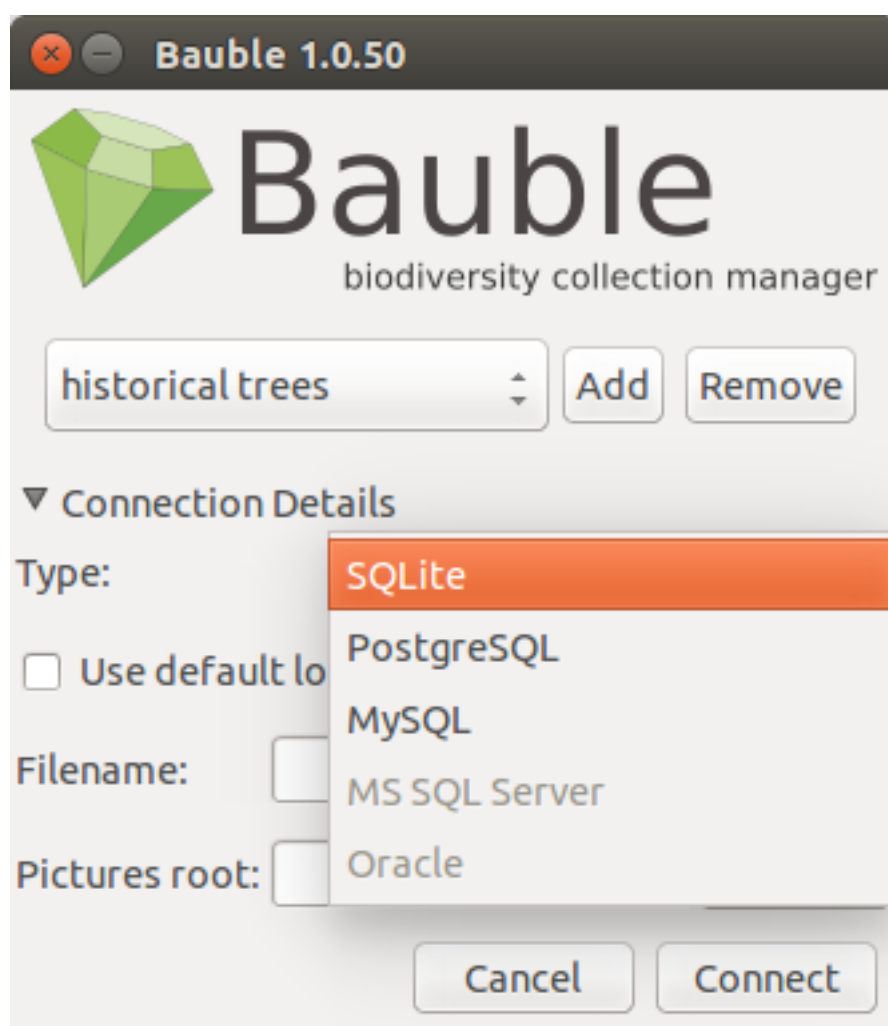
Si no sabría que hacer acá, Bauble le da una mano. Active el **Utilice los valores por defecto** y finalice el crear la conexión pisando **Conectar**.

El resto de esta sección da más detalles de natura técnica. Si no quiere adentrarse, puede seguir con la sección siguiente.

detalles de conexión: sintonización fina

By default Bauble uses the file-based SQLite database. During the installation process you had the choice (and you still have after installation), to add database connectors other than the default SQLite.

In this example, Bauble can connect to SQLite, PostgreSQL and MySQL, but no connector is available for Oracle or MS SQL Server.



If you use SQLite, all you really need specify is the connection name. If you let Bauble use the default filename then Bauble creates a database file with the same name as the connection and .db extension, and a pictures folder with the same name and no extension, both in `~/ .bauble` on Linux/MacOSX or in `AppData\Roaming\Bauble` on Windows.

Still with SQLite, you might have received or downloaded a bauble database, and you want to connect to it. In this case you do not let Bauble use the default filename, but you browse in your computer to the location where you saved the Bauble SQLite database file.

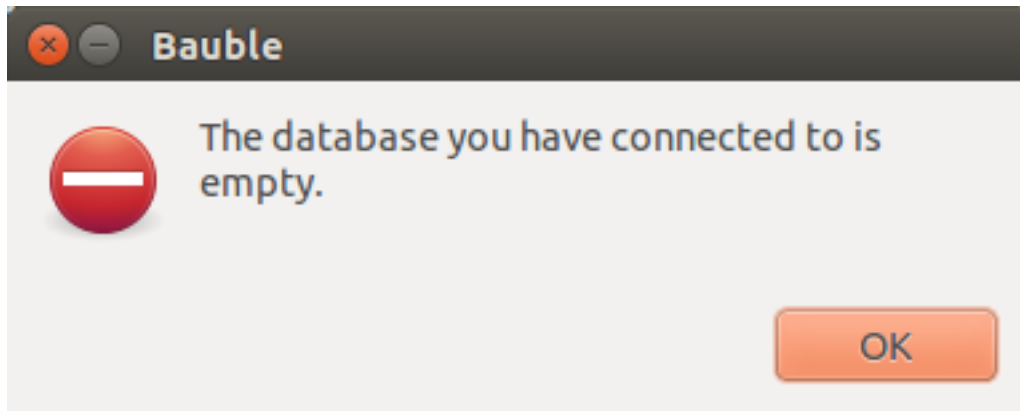
If you use a different database connector, the dialog box will look different and it will offer you the option to fine tune all parameters needed to connect to the database of your choice.

If you are connecting to an existing database you can continue to *Editar e Insertar información* and subsequently *Searching in Bauble*, otherwise read on to the following section on initializing a database for Bauble.

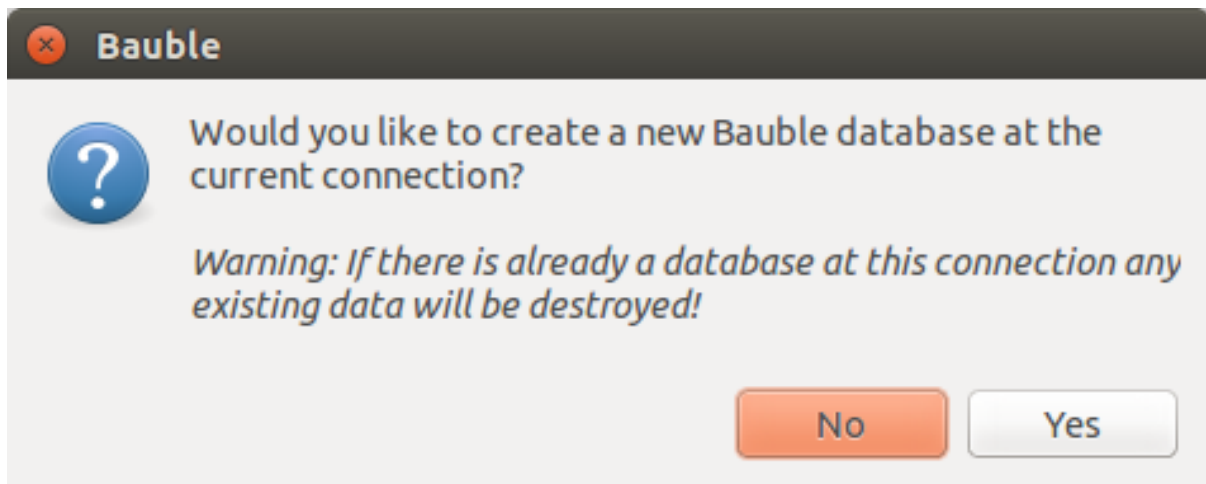
If you plan to associate pictures to plants, specify also the *pictures root* folder. The meaning of this is explained in further detail at *Fotos* in *Editar e Insertar información*.

3.1.3 Inicializar una base de datos

La primera vez que se abre una conexión a una base de datos, nunca antes vista con Bauble, el programa presenta el aviso:



e inmediatamente después la pregunta:



Be careful when manually specifying the connection parameters: the values you have entered may refer to an existing database, not intended for use with Bauble. By letting Bauble initialize a database, the database will be emptied and all of its content be lost.

If you are sure you want to create a database at this connection then select «Yes». Bauble will then start creating the database tables and importing the default data. This can take a minute or two so while all of the default data is imported into the database so be patient.

Cuando se haya creado, configurado, inicializado la base de datos, estamos listos para [Editar e Insertar información](#) y lo más interesante [Searching in Bauble](#).

3.2 Searching in Bauble

Searching allows you to view, browse and create reports from your data. You can perform searches by either entering the queries in the main search entry or by using the Query Builder to create the queries for you. The results of Bauble searches are listed in the main window.

3.2.1 Search Strategies

There are three types of search strategies available in Bauble. Considering the search strategies available in Bauble, sorted in increasing complexity: you can search by value, expression or query.

Searching by query, the most complex and powerful, is assisted by the Query Builder, described below.

Todas búsquedas son sin contar mayúsculas o minúsculas, así que se busque Maxillaria o maxillaria, se habrán los mismos resultados.

Search by Value

Search by value is the simplest way to search. You just type in a string and see what matches. Which fields/columns are searched for your string depends on how the different plugins are configured. For example, by default the PlantPlugin searches the family name, the genus name, the species and infraspecific species names, vernacular names and geography. So if you want to search in the notes field of any of these types then searching by value is not the search you're looking for.

Examples of searching by value would be: Maxillaria, Acanth, 2008.1234, 2003.2.1

Search strings are separated by spaces. For example if you enter the search string `Block 10` then Bauble will search for the strings `Block` and `10` and return all the results that match either of these strings. If you want to search for `Block 10` as a whole string then you should quote the string like `"Block 10"`.

Search by Expression

Searching with expression gives you a little more control over what you are searching for. It can narrow the search down to a specific domain. Expression consists of a domain, an operator and a value. For example the search: `gen=Maxillaria` would return all the genera that match the name `Maxillaria`. In this case the domain is `gen`, the operator is `=` and the value is `Maxillaria`.

The search string `gen like max%` would return all the genera whose names start with «Max». In this case the domain again is `gen`, the operator is `like`, which allows for «fuzzy» searching and the value is `max%`. The percent sign is used as a wild card so if you search for `max%` then it searches for all values that start with `max`. If you search for `%max` it searches for all values that end in `max`. The string `%max%a` would search for all values that contain `max` and end in `a`.

For more information about the different search domains and their short-hand aliases, see [search-domains](#).

If expressions are invalid they are usually used as search by value searches. For example the search string `gen=` will execute a search by value for the string `gen` and the search string `gen like` will search for the string `gen` and the string `like`.

Search by Query

Queries allow the most control over searching. With queries you can search across relations, specific columns and join search using boolean operators like AND and OR.

An example of a query would be:

```
plant where accession.species.genus.family=Fabaceae and location.
↪site="Block 10"
```

This query would return all the plants whose family are Fabaceae and are located in Block 10.

Searching with queries usually requires some knowledge of the Bauble internals and database table layouts.

Un par de ejemplos útiles:

- Which locations are in use:

```
location where plants.id!=0
```

- Which genera are associated to at least one accession:

```
genus where species.accession.id!=0
```

Domains

The following are the common search domain and the columns they search by default. The default columns are used when searching by value and expression. The queries do not use the default columns.

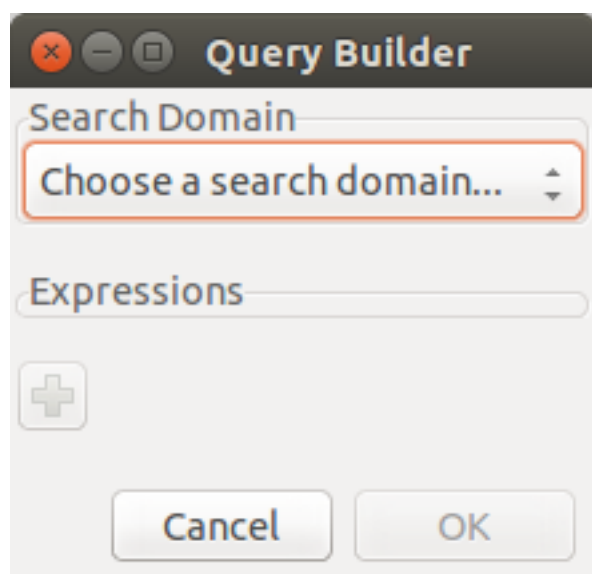
Domains family, fam: Search `bauble.plugins.plants.Family`
 genus, gen: Search `bauble.plugins.plants.Genus`
 species, sp: Search `bauble.plugins.plants.Species`
 geography: Search `bauble.plugins.plants.Geography`
 acc: Search `bauble.plugins.garden.Accession`
 plant: Search `bauble.plugins.garden.Plant`
 location, loc: Search `bauble.plugins.garden.Location`

3.2.2 The Query Builder

The Query Builder helps you build complex search queries through a point and click interface. To open the Query Builder click the to the left of the search entry or select *Tools→Query Builder* from the menu.

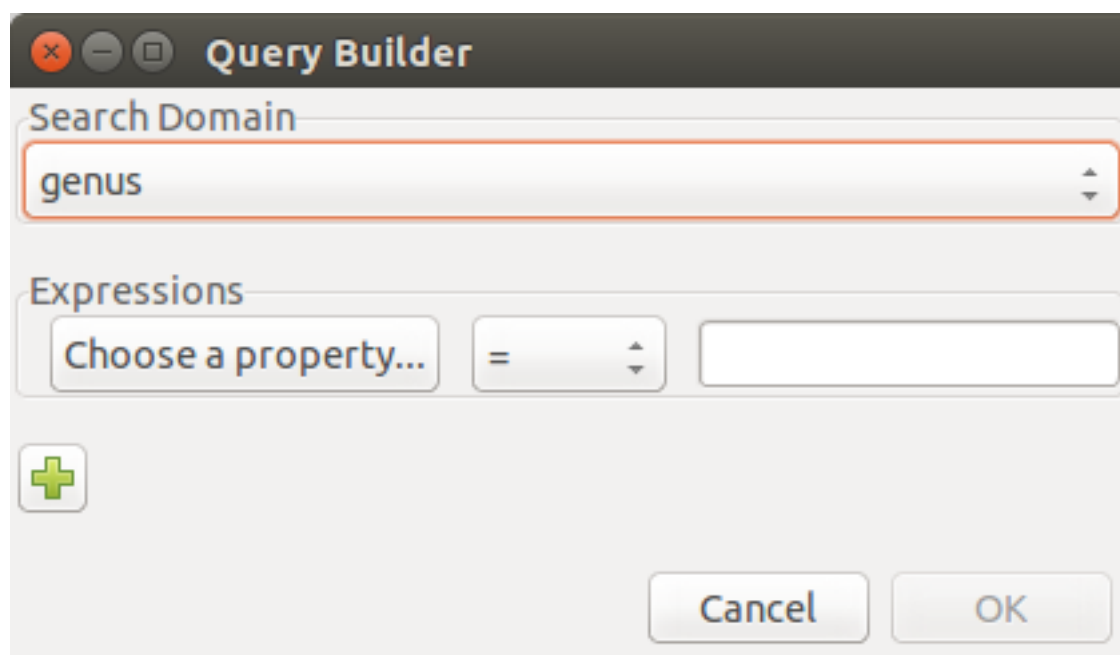
The Query Builder composes a query that will be understood by the Query Search Strategy described above. You can use the Query Builder to get a feeling of correct queries before you start typing them by hand, something that you might prefer if you are a fast typer.

After opening the Query Builder you must select a search domain. The search domain will determine the type of data that is returned and the properties that you can search.



The search domain is similar to a table in the database and the properties would be the columns on the table. Often the table/domain and properties/columns are the same but not always.

Once a search domain is selected you can then select a property of the domain to compare values to. The search operator can then be changed for how you want to make the search comparison. Finally you must enter a value to compare to the search property.



If the search property you have selected can only have specific values then a list of possible values will be provided for you to choose from.

If multiple search properties are necessary then clicking on the plus sign will add more search properties. Select And/Or next to the property name choose how the properties will be combined in the search query.

When you are done building your query click OK to perform the search.

3.3 Editar e Insertar información

The main way that we add or change information in Bauble is by using the editors. Each basic type of data has its own editor. For example there is a Family editor, a Genus editor, an Accession editor, etc.

To create a new record click on the *Insert* menu on the menubar and then select the type of record you would like to create. This will open a new blank editor for the type.

To edit an existing record in the database right click on an item in the search results and select *Edit* from the popup menu. This will open an editor that will allow you to change the values on the record that you selected.

Most types also have children which you can add by right clicking on the parent and selecting «Add ???...» on the context menu. For example, a Family has Genus children: you can add a Genus to a Family by right clicking on a Family and selecting «Add genus».

3.3.1 Notas

Aproximadamente todos los editores en Bauble tienen una pestaña *Notas*, que debería funcionar del mismo modo, independientemente del editor empleado.

If you enter a web address in a note then the link will show up in the Links box when the item you are editing is selected in the search results.

You can browse the notes for an item in the database using the Notes box at the bottom of the screen. The Notes box will be desensitized if the selected item does not have any notes.

3.3.2 Familia

The Family editor allows you to add or change a botanical family.

The *Family* field on the editor will change the name of the family. The Family field is required.

The *Qualifier* field will change the family qualifier. The value can either be *sensu lato*, *sensu stricto* or nothing.

Synonyms allow you to add other families that are synonyms with the family you are currently editing. To add a new synonym type in a family name in the entry. You must select a family name from the list of completions. Once you have selected a family name that you want to add as a synonym click on the Add button next to the synonym list and it will add the selected synonym to the list. To remove a synonym select the synonym from the list and click on the Remove button.

To cancel your changes without saving then click on the *Cancel* button.

To save the family you are working on then click *OK*.

To save the family you are working on and add a genus to it then click on the *Add Genera* button.

To add another family when you are finished editing the current one click on the *Next* button on the bottom. This will save the current family and open a new blank family editor.

3.3.3 Género

The Genus editor allows you to add or change a botanical genus.

The *Family* field on the genus editor allows you to choose the family for the genus. When you begin type a family name it will show a list of families to choose from. The family name must already exist in the database before you can set it as the family for the genus.

The *Genus* field allows you to set the genus for this entry.

The *Author* field allows you to set the name or abbreviation of the author(s) for the genus.

Synonyms allow you to add other genera that are synonyms with the genus you are currently editing. To add a new synonyms type in a genus name in the entry. You must select a genus name from the list of completions. Once you have selected a genus name that you want to add as a synonym click on the Add button next to the synonym list and it will add the selected synonym to the list. To remove a synonym select the synonym from the list and click on the Remove button.

To cancel your changes without saving then click on the *Cancel* button.

To save the genus you are working on then click *OK*.

To save the genus you are working on and add a species to it then click on the *Add Species* button.

To add another genus when you are finished editing the current one click on the *Next* button on the bottom. This will save the current genus and open a new blank genus editor.

3.3.4 Species/Taxon

For historical reasons called a *species*, but by this we mean a *taxon* at rank *species* or lower. It represents a unique name in the database. The species editor will allow you to construct the name as well as associate metadata with the taxon such as its distribution, synonyms and other information.

The *Infraspecific parts* in the species editor will allow you to specify the *taxon* further than at *species* rank.

To cancel your changes without saving then click on the *Cancel* button.

To save the species you are working on then click *OK*.

To save the species you are working on and add an accession to it then click on the *Add Accession* button.

To add another species when you are finished editing the current one click on the *Next* button on the bottom. This will save the current species and open a new blank species editor.

3.3.5 Accesiones

The Accession editor allows us to add an accession to a species. In Bauble an accession represents a group of plants or clones. The accession would refer maybe a group of seed or cuttings from a species. A plant would be an individual from that accession, i.e. a specific plant in a specific location.

Fuente de la accesión

The source of the accessions lets you add more information about where this accession came from. At the moment the type of the source can be either a Collection or a Donation.

Colecta

Una colección.

Donation

Una donación.

3.3.6 Planta

The Plant editor.

Creating multiple plants

You can create multiple Plants by using ranges in the code entry. This is only allowed when creating new plants and it is not possible when editing existing Plants in the database.

For example the range, 3-5 will create plant with code 3,4,5. The range 1,4-7,25 will create plants with codes 1,4,5,6,7,25.

When you enter the range in the plant code entry the entry will turn blue to indicate that you are now creating multiple plants. Any fields that are set while in this mode will be copied to all the plants that are created.

Fotos

Just as almost all objects in the Bauble database can have *Notes* associated to them, Plants can have *Pictures*: next to the tab for Notes, the Plants editor contains an extra tab called «Pictures». You can associate as many pictures as you might need to a plant.

When you associate a picture to a plant, the file is copied in the *pictures* folder, and a miniature (500x500) is generated and copied in the *thumbnails* folder inside of the pictures folder.

Al momento (Bauble 1.0.41), las imagenes no son guardadas en la base de datos. para que las fotos sean disponibles en todos los terminales en que se quiera utilizar Bauble, será necesario instalar un servicio de condivisión de archivos, como pueden ser Copy o Dropbox. Quien escribe esta documentación sugiere utilizar Copy porque ofrece más espacio y por su política «Fair Storage».

Remember that you have configured the pictures root folder when you specified the details of your database connection. Again, you should make sure that the pictures root folder is shared with your file sharing service of choice.

When a Plant in the current selection is highlighted, its pictures are displayed in the pictures pane, the pane left of the information pane. When an accession in the selection is highlighted, any picture associated to the plants in the highlighted accession are displayed in the pictures pane.

3.3.7 Locations

The Location editor

danger zone

The location editor contains an initially hidden section named *danger zone*. The widgets contained in this section allow the user to merge the current location into a different location, letting the user correct spelling mistakes or implement policy changes.

3.4 Tagging

Tagging is an easy way to give context to an object or create a collection of object that you want to recall later. For example if you want to collect a bunch of plants that you later want to create a report from you can tag them with the string «for that report i was thinking about». You can then select «for that report i was thinking about» from the tags menu to show you all the objects you tagged.

Tagging can be done two ways. By selecting one or more items in the search results and pressing Ctrl-T or by selecting *Tag*→*Tag Selection* from the menu. If you have selected multiple items then only that tags that are common to all the selected items will have a check next to it.

3.5 generando reportes

3.5.1 el uso del reporte del formateador Mako

The Mako report formatter uses the Mako template language for generating reports. More information about Mako and its language can be found at makotemplates.org.

The Mako templating system should already be installed on your computer if Bauble is installed.

Creating reports with Mako is similar in the way that you would create a web page from a template. It is much simpler than the XSL Formatter(see below) and should be relatively easy to create template for anyone with a little but of programming experience.

The template generator will use the same file extension as the template which should indicate the type of output the template with create. For example, to generate an HTML page from your template you should name the template something like *report.html*. If the template will generate a comma seperated value file you should name the template *report.csv*.

The template will receive a variable called *values* which will contain the list of values in the current search.

The type of each value in *values* will be the same as the search domain used in the search query. For more information on search domains see [Domains](#).

If the query does not have a search domain then the values could all be of a different type and the Mako template should prepared to handle them.

3.5.2 Using the XSL Report Formatter

The XSL report formatter requires an XSL to PDF renderer to convert the data to a PDF file. Apache FOP is is a free and open-source XSL->PDF renderer and is recommended.

If using Linux, Apache FOP should be installable using your package manager. On Debian/Ubuntu it is installable as `fop` in Synaptic or using the following command:

```
apt-get install fop
```

Installing Apache FOP on Windows

You have two options for installing FOP on Windows. The easiest way is to download the prebuilt [ApacheFOP-0.95-1-setup.exe](#) installer.

Alternatively you can download the [archive](#). After extracting the archive you must add the directory you extracted the archive to to your PATH environment variable.

3.6 Importing and Exporting Data

Aunque puede extenderse Bauble mediante plugins para que admita formatos alternativos de importación y exportación, por defecto solo importa/exporta ficheros con datos separados por coma o CSV.

There is some support for exporting to the Access for Biological Collections Data it is limited.

There is also limited support for exporting to an XML format that more or less reflects exactly the tables and row of the database.

Exporting ABCD and XML will not be covered here.

Advertencia: Importing files will most likely destroy any data you have in the database so make sure you have backed up your data.

3.6.1 Importing from CSV

In general it is best to only import CSV files into Bauble that were previously exported from Bauble. It is possible to import any CSV file but that is more advanced than this doc will cover.

To import CSV files into Bauble select *Tools*→*Export*→*Comma Separated Values* from the menu.

After clicking OK on the dialog that asks if you are sure you know what you're doing a file chooser will open. In the file chooser select the files you want to import.

3.6.2 Exporting to CSV

To export the Bauble data to CSV select *Tools*→*Export*→*Comma Separated Values* from the menu.

This tool will ask you to select a directory to export the CSV data. All of the tables in Bauble will be exported to files in the format *tablename.txt* where *tablename* is the name of the table where the data was exported from.

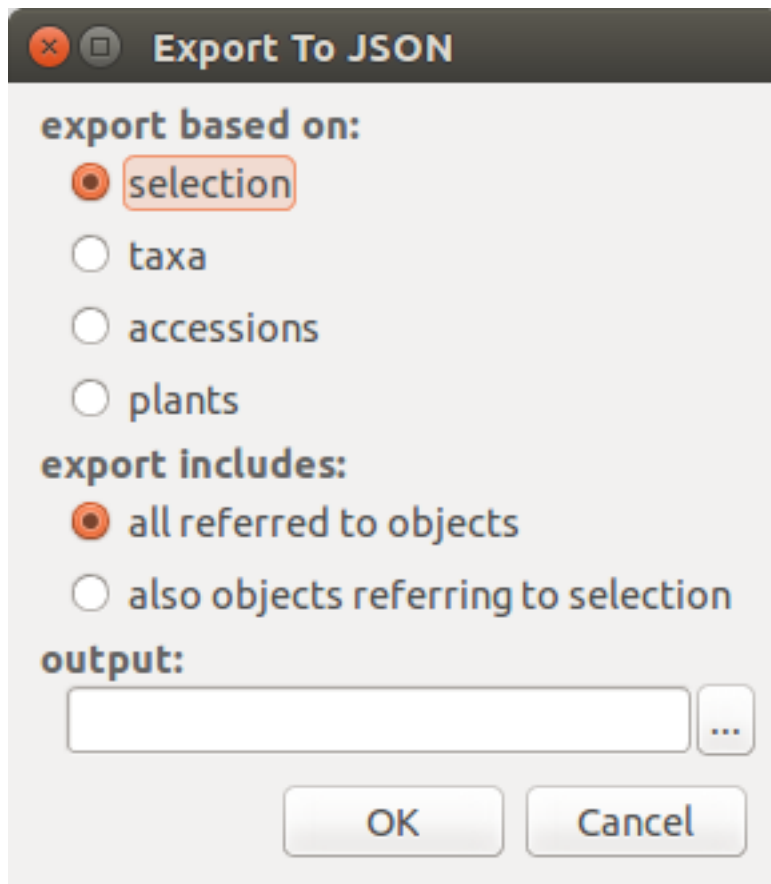
3.6.3 Importing from JSON

This is *the* way to import data into an existing database, without destroying previous content. A typical example of this functionality would be importing your digital collection into a fresh, just initialized Bauble database. Converting a database into bauble json interchange format is beyond the scope of this manual, please contact one of the authors if you need any further help.

Using the Bauble json interchange format, you can import data which you have exported from a different Bauble installation.

3.6.4 Exporting to JSON

This feature is still under development.



when you activate this export tool, you are given the choice to specify what to export. You can use the current selection to limit the span of the export, or you can start at the complete content of a domain, to be chosen among Species, Accession, Plant.

Exporting *Species* will only export the complete taxonomic information in your database. *Accession* will export all your accessions plus all the taxonomic information it refers to: unreferred to taxa will not be exported. *Plant* will export all living plants (some accession might not be included), all referred to locations and taxa.

3.7 Managing Users

Nota: The Bauble users plugin is only available on PostgreSQL based databases.

The Bauble User's Plugin will allow you to create and manage the permissions of users for your Bauble database.

3.7.1 Creating Users

To create a new user. . .

3.7.2 Permissions

Bauble permite la lectura, escritura y ejecución.

CAPÍTULO 4

Administración

4.1 Administración

Si está utilizando un verdadero sistema de manejo de base de datos (un DBMS) para guardar sus datos, es preciso saber algo de como se administra una base de datos. Esto está muy fuera del alcance de la documentación de Bauble, sin embargo es importante que los usuarios sean conscientes de la necesidad.

4.1.1 SQLite

De SQLite no se puede decir que sea un verdadero DBMS: cada database SQLite no es más que un archivo. De estos archivo haganse copias de seguridad y estará bien. Si no sabe donde buscar los archivos SQLite representantes sus colecciones, considere que, por defecto, Bauble pone sus datos en la carpeta `~/bauble` (en Windows estaría en algún lugar en su carpeta `AppData`).

4.1.2 MySQL

Por favor utilice la documentación oficial.

4.1.3 PostgreSQL

Por favor utilice la documentación oficial. Una discusión muy detallada sobre las opciones de backup empieza al [chapter_24](#).

5.1 Downloading the source

The Bauble source can be downloaded from our source repository on [github](#).

If you want a particular version of Bauble, we release and maintain versions into branches. you should `git checkout` the branch corresponding to the version of your choice. Branch names for Bauble versions are of the form `bauble-x.y`, where `x.y` can be `1.0`, for example. Our workflow is to commit to the *master* development branch or to a *patch* branch and to include the commits into a *release* branch when ready.

To check out the most recent code from the source repository you will need to install the [Git](#) version control system. Git is included in all reasonable Linux distributions and can be installed on all current operating systems.

Once you have installed Git you can checkout the latest Bauble code with the following command:

```
git clone https://github.com/Bauble/bauble.classic.git
```

For more information about other available code branches go to [bauble.classic on github](#).

5.2 Developer's Manual

5.2.1 helping bauble development

Installing Bauble always includes downloading the sources, connected to the github repository. This is so because in our eyes, every user is always potentially also a developer.

If you want to contribute to Bauble, you can do so in quite a few different ways:

```
* use the software, note the things you don't like, open issue for_
→each of them. a developer will react.
* if you have an idea of what you miss in the software but can't_
→quite
  formalize it into separate issues, you could consider hiring a
  professional. this is the best way to make sure that something_
→happens
  quickly on Bauble. do make sure the developer opens issues and_
→publishes
  their contribution on github.
* translate! any help with translations will be welcome, so please_
→do! you
  can do this without installing anything on your computer, just_
→using the
  on-line translation service offered by http://hosted.weblate.org/
* fork the repository, choose an issue, solve it, open a pull_
→request. see
  the `bug solving workflow`_ below.
```

5.2.2 bug solving workflow

normal development workflow

- while using the software, you notice a problem, or you get an idea of something that could be better, you think about it good enough in order to have a very clear idea of what it really is, that you noticed. you open an issue and describe the problem. someone might react with hints.
- you open the issues site and choose one you want to tackle.
- assign the issue to yourself, this way you are informing the world that you have the intention to work at it. someone might react with hints.
- optionally fork the repository in your account and preferably create a branch, clearly associated to the issue.
- write unit tests and commit them to your branch (do not commit failing unit tests to the `master` branch).
- write more unit tests (ideally, the tests form the complete description of the feature you are adding or correcting).
- make sure the feature you are adding or correcting is really completely described by the unit tests you wrote.
- make sure your unit tests are atomic, that is, that you test variations on changes along one single variable. do not give complex input to unit tests or tests that do not fit on one screen (25 lines of code).
- write the code that makes your tests succeed.

- update the `il8n` files (run `./scripts/il8n.sh`).
- whenever possible, translate the new strings you put in code or glade files.
- commit your changes.
- push to github.
- open a pull request.

publishing to production

- open the pull request page using as base the production line, compared to `master`.
- make sure a `bump` commit is included in the differences.
- it should be possible to automatically merge the branches.
- create the new pull request, call it as “publish to the production line”.
- you possibly need wait for travis-ci to perform the checks.
- merge the changes.
- tell the world about it: on facebook, the google group, linkedin, ...

closing step

- review this workflow. consider this as a guideline, to yourself and to your colleagues. please help make it better and matching the practice.

5.2.3 structure of user interface

the user interface is built according to the Model-View-Presenter architectural pattern. The **view** is described in a glade file and is totally dumb, you do not subclass it because it implements no behaviour and because its appearance is, as said, described elsewhere, including the association signal-callbacks. The **model** simply follows the sqlalchemy practices.

You will subclass the **presenter** in order to:

- define `widget_to_field_map`, the association from name of view object to name of model attribute,
- override `view_accept_buttons`, the list of widget names which, if activated by the user, mean that the view should be closed,
- define all needed callbacks,

The presenter should not know of the internal structure of the view, instead, it should use the view api to set and query the values inserted by the user. The base class for the presenter, `GenericEditorPresenter` defined in `bauble.editor`, implements many generic callbacks.

Model and Presenter can be unit tested, not the View.

The `Tag` plugin is a good minimal example, even if the `TagItemGUI` falls outside this description. Other plugins do not respect the description.

We use the same architectural pattern for non-database interaction, by setting the presenter also as model. We do this, for example, for the JSON export dialog box.

5.2.4 building (on Windows)

Building a python program is a bit of a contraddiction. You don't normally *build* nor *compile* a python program, you run it in its (virtual) environment, and python will process the modules loaded and produce faster-loading *compiled* python files. You can, however, produce a *Windows executable* from a python script, executable containing the whole python environment and dependencies.

1. Follow all steps needed to set up a working Bauble environment from [Instalación](#), but skip the final `install` step.
2. instead of *installing* Bauble, you produce a Windows executable. This is achieved with the `py2exe` target, which is only available on Windows systems:

```
python setup.py py2exe
```

3. At this point you can run Bauble. To run the compiled executable run:

```
.\dist\bauble.exe
```

or copy the executable to wherever you think appropriate.

4. To optionally build an NSIS installer package you must install NSIS from nsis.sourceforge.net. After installing NSIS right click on `.\scripts\build.nsi` in Explorer and select *Compile NSIS Script*.

5.3 Extending Bauble with Plugins

Nearly everything about Bauble is extensible through plugins. Plugins can create tables, define custom searches, add menu items, create custom commands and more.

To create a new plugin you must extend the `bauble.pluginmgr.Plugin` class.

5.4 API Documentation

5.4.1 bauble

5.4.2 bauble.db

`bauble.db.Base`

All tables/mappers in Bauble which use the SQLAlchemy declarative plugin for decla-

ring tables and mappers should derive from this class.

An instance of `sqlalchemy.ext.declarative.Base`

`bauble.db.metadata`

The default metadata for all Bauble tables.

An instance of `sqlalchemy.schema.MetaData`

5.4.3 `bauble.connmgr`

5.4.4 `bauble.editor`

5.4.5 `bauble.i18n`

5.4.6 `bauble.ui`

5.4.7 `bauble.meta`

5.4.8 `bauble.paths`

5.4.9 `bauble.pluginmgr`

5.4.10 `bauble.prefs`

5.4.11 `bauble.task`

5.4.12 `bauble.types`

5.4.13 `bauble.utils`

5.4.14 `bauble.view`

class `bauble.view.SearchView.ViewMeta`

5.4.15 `bauble.search`

5.4.16 `bauble.plugins.plants`

5.4.17 `bauble.plugins.garden`

5.4.18 `bauble.plugins.abcd`

5.4.19 `bauble.plugins.imex`

5.4.20 `bauble.plugins.report`

5.4.21 `bauble.plugins.report.xsl`

5.4.22 `bauble.plugins.report.mako`

5.4.23 `bauble.plugins.tag`

CAPÍTULO 6

Apoyar el proyecto Bauble

Si usted utiliza Bauble, o si le parece poder apoyar su desarrollo, por favor considere una [donación](#)

B

bauble.db.Base (variable incorporada), [36](#)

bauble.db.metadata (variable incorporada), [37](#)

bauble.view.SearchView.ViewMeta (clase incorporada), [37](#)