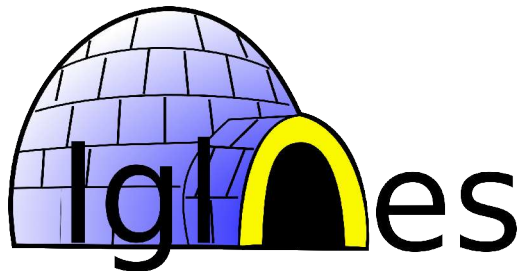


Unificación de criterios

Guía de estilo para el desarrollo de BulmaGés

AUTOR

Este artículo puede ser copiado y distribuido en las condiciones de la licencia GNU
para documentación libre, GFDL (<http://www.gnu.org/copyleft/fdl.html>)



[Implantación de Gestión Libre y Universal para Empresas y Sociedades]

Unificación de criterios

Guía de estilo para el desarrollo de BulmaGés

AUTOR

Esta guía de estilo es un resumen de algunas normas no escritas que voy aprendiendo mientras voy programando y consultando a los otros desarrolladores del proyecto. Quizás cuando esta guía crezca deberán agruparse los puntos de forma lógica.

1. Sobre nomenclatura propia.

1. Entendemos que un listado es una consulta de los datos de una tabla que se muestran visualmente en forma de grid o tabla. Nos referiremos a los tradicionales listados en papel o formato digital como informes.

2. Sobre nomenclatura de variables, miembros (funciones) y clases.

1. En cada nuevo programa que uno haga (ya sea desde cero, como si se copia otro fuente para crear el nuevo programa) el autor debe poner su nombre. Esto se hace para garantizar la continuidad del proyecto bajo licencia GPL. Ejemplo del texto inicial del código fuente de un programa (provedit.cpp):

```
/*****  
 *   Copyright (C) 2004 by J. M. Torres Rigo   *  
*****/
```

```

* joanmi@bulma.net *
*
* This program is free software; 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. *
*
* This program is distributed in the hope that it will be useful, *
* but WITHOUT ANY WARRANTY; without even the implied warranty of *
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *
* GNU General Public License for more details. *
*
* You should have received a copy of the GNU General Public License *
* along with this program; if not, write to the *
* Free Software Foundation, Inc., *
* 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA. *
*****/

```

3. Todo en C++. El lenguaje de programación es el C++.
4. Los nombres de las clases, siempre empiezan en mayúsculas.
5. Los nombres de las clases se escriben todos en minúscula. Por ejemplo orders-listbase.
6. Los nombres de las clases, miembros (funciones) y variables se escriben en inglés. El motivo es la idea de que el proyecto algún día pueda integrarse en KDE.
7. Cuando es una palabra compuesta se juntan y se pone la primera letra de cada palabra en mayúsculas y el resto en minúsculas (incluida la primera letra del nombre de la función) por ejemplo: `int elementName() const { return m_elementName; };`. Creo que `m_` viene de member. La idea es poder distinguir rápidamente las variables de una clase y las locales de la función que estás leyendo que nunca llevan `m_`.
8. No se pone nunca "get" para indicar que es una función que devuelve un valor sino simplemente el nombre de la propiedad que devuelve. En vez de `getVa-`

lue/setValue o get_value/set_value nosotros usamos sólo value/setValue.

9. En la lectura de tablas mediante cursores se utilizará la clase propia "cursor2".
10. Los nombres de los iconos/imágenes empezaran con el prefijo "i_".

3. Sobre la interfaz gráfica.

1. Las pantallas en las que se pueda modificar la información en la base de datos deberán haber dos botones uno de aceptar y otro cancelar (aceptar a la derecha y cancelar a la izquierda).
2. En alguna pantalla he visto que los slots de las pantallas empiezan por A mayuscula quizás deberíamos seguir una nomenclatura al estilo de "a_" para ser coherentes con la nomenclatura de los campos.
3. Desde un listado (una QTable) para llamar a la consulta de ese campo pondremos el carácter "+". Por ejemplo, en la consulta de pedidos (linorderslist.cpp) cuando en la columna de artículo cuando se quiere consultar artículos escribimos "+" y al salir de la celda nos abre el listado de artículos.
4. Desde un listado (una QTable) para copiar un dato de la línea superior pondremos el carácter "*". Por ejemplo, en la consulta de pedidos (linorderslist.cpp) cuando en la columna de cantidad se quiere copiar el dato de la línea anterior escribimos "*" y al salir de la celda nos ha copiado el valor de la línea anterior.
5. Para distinguir en el código si una función (miembro) es un slot que se usa en un widget del formulario, los slots deberían distinguirse con un prefijo, que puede ser s_.

4. Sobre los comentarios en los fuentes.

1. Ver la documentación sobre Dxygen en la web (www.iglu.es) o bien en el cvs (bulimages/iglu.es/manuales/doxmanual) de David Gutiérrez.
2. En los listados incluiremos en el mismo fuente como documentación la definición de la tabla principal de la que se extrae la información del listado. Ver

como ejemplo el fuente "delivnoteslis.cpp".

5. Sobre la base de datos.

1. Los nombres de tabla se escriben en singular y los nombres de sus campos se componen de un prefijo significativo y como sufijo el nombre de la tabla. Por ejemplo, tabla "proveedor", campo identificador de proveedor "idproveedor".
2. Los campos tipo autonuméricos que identifican un registro de una tabla se llamarán id+nombre_de_tabla.

6. Sobre los nombres de los fuentes.

1. Los nombres de los ficheros de pantallas (*.ui) se acabarán con el sufijo base. (hay otro sufijo pero ahora no recuerdo cual es y como se usa, comentar con Tomeu o ver en la documentación de kde).
2. Los nombres de los fuentes de programas de listado se acabarán con el sufijo "list". Por ejemplo orderslist.cpp (listado de pedidos).
3. Los nombres de los fuentes de los programas de pantallas tipo "ficha de..." o de edición tomarán el nombre de la tabla de la que cargan sus datos.