VELIGER - Editor de metadados Documentation

Versão 0.6.0b

Bruno C. Vellutini

24/05/2010

Conteúdo

VÉLIGER é o editor de metadados do banco de imagens do Centro de Biologia Marinha da Universidade de São Paulo (CEBIMar-USP).

Este programa abre imagens JPG, lê seus metadados (IPTC) e fornece uma interface para editar estas informações. Os campos foram adaptados para o propósito do banco, que é divulgar imagens relacionadas com biologia marinha.

Os campos editáveis são:

IPTC título, legenda, marcadores, táxon, espécie, especialista, autor, direitos, tamanho, local, cidade, estado e país.

EXIF Geolocalização.

CAPÍTULO 1

Guia rápido

Instale o programa seguindo estas instruções (link) e inicie-o.

Ao importar imagens o programa irá ler os metadados de cada uma e criar as respectivas entradas na tabela principal. Selecione uma linha da tabela para visualizar o thumbnail e os metadados contidos na imagem.

Para editar clique 2x na célula da tabela ou clique e aperte F2; após digitar aperte *Enter* e o valor estará salvo no programa (mas ainda não foi gravado na imagem).

Outra maneira de editar é usando os campos de edição. Após selecionar uma entrada na tabela os campos de edição serão preenchidos com os metadados da imagem. Para editar simplesmente escreva as informações nos campos desejados e aperte **ctrl+s** [1] antes de mudar para outra imagem. Caso você mude a linha selecionada antes de salvar as alterações serão perdidas.

Nota: Lembre-se que salvar com ctrl+s, ou editando direto na tabela, não grava as alterações no arquivo da imagem. Os novos metadados ficam salvos apenas no programa, mas de maneira persistente (não é perdido após reiniciá-lo).

Após a alteração dos metadados a entrada é adicionada à uma lista que mostra os arquivos cujos metadados foram modificados, mas ainda não foram gravados no arquivo (ou seja, o arquivo ainda está com os metadados originais). Para gravar os metadados na imagem utilize o atalho **ctrl+shift+s** [2]. Se tudo correr bem na gravação os novos metadados estarão embebidos na imagem.

Nota: A codificação utilizada para gravar os metadados é UTF-8. Veja mais informações sobre a codificação dos metadados abaixo.

Para editar vários valores de uma coluna, selecione a célula de uma entrada e arraste até outra entrada na mesma coluna (ou clique e segure **shift** ou **ctrl** para selecionar outras); com as entradas selecionadas aperte F2 e ele vai abrir o campinho de edição na tabela; escreva e aperte *Enter* e os valores serão copiados para as entradas selecionadas (apenas para a coluna escolhida).

Para alterar todos os metadados de várias entradas de uma vez selecione as entradas que serão modificadas e preencha os campos de edição. Ao apertar **ctrl+s** todas as entradas selecionadas terão seus metadados sobrescritos pelo que você tiver inserido nos campos de edição.

Também é possível copiar os metadados de uma entrada e colá-los em outra(s). Para tal, basta selecionar uma entrada e apertar **ctrl+c** para copiar, selecionar as entradas cujos metadados serão sobrescritos e apertar **ctrl+v**.

Nota: Lembre-se que ao salvar os metadados (**ctrl+s**) todos os metadados das entradas selecionadas serão **sobrescritos**, mesmo você tendo modificado apenas um dos campos (a legenda, por exemplo). O mesmo ocorrerá ao colar

[1] ou use o ícone para salvar no menu de ferramentas.
[2] ou clique em gravar na doca de modificadas.

os metadados copiados de uma entrada em outras usando **ctrl+c/ctrl+v**. O resultado destas duas funções são entradas cujos metadados são idênticos.

Ao fechar o programa ele lembra a tabela e suas modificações, portanto pode fechar sem apertar **Ctrl+Shift+S** que as modificações continuaram presentes quando programa for aberto novamente.

1.1 Geolocalização

Ao clicar numa entrada, se a doca da geolocalização estiver aberta, o mapa e as coordenadas gravadas na imagem serão carregadas. Para mudar a localização da imagem simplesmente arraste o marcador no mapa até a nova posição, aperte o botão **Atualizar** para carregar a nova localização no editor e aperte **Gravar** para salvar as alterações na imagem.

Se múltiplas imagens estiverem selecionadas, a geolocalização será gravada em todas.

Nota: É necessário apertar **Atualizar** após arrastar. As novas coordenadas não são atualizadas automaticamente (ainda).

Imagens sem coordenadas gravadas no EXIF mostrarão um mapa sem marcador. Para selecionar um local clique com o **botão direito** do mouse no mapa [3]. O zoom será aumentado para que você possa refinar a posição arrastando o marcador. Lembre-se de atualizar antes de gravar.

Nota: Existe um bug no API do Google Maps ou no WebKit que faz com que o marcador seja colocado na posição errada. Isso ocorre quando o usuário arrasta o mapa ou aumenta o zoom, mudando a posição central inicial, antes de criar o marcador. Por este motivo, ao criar um novo marcador tente clicar na região desejada primeiro para evitar contratempos. Após criá-lo não haverá problemas para arrastar até a posição específica.

1.2 Conversão da codificação

Se você observar erros na codificação de caracteres especiais (ç, à, á, ã, ê) dos metadados após importar imagens, o programa que inseriu estes dados deve ter utilizado o padrão Latin-1 (ISO 8859-1). Para corrigir a codificação para o padrão preferido atualmente, o UTF-8, utilize a função de conversão no menu Editar. Essa alteração é gravada na imagem, logo, tenha backup sempre, pois se você voltar a editar esta imagem com outro programa pode ser que ele não reconheça o UTF-8 corretamente (acho difícil, mas pode acontecer).

1.3 Docas

A janela principal do **VÉLIGER** contém 4 docas: editor dos metadados, editor da geolocalização, miniatura da imagem e lista de entradas modificadas. Estas docas tem um posicionamento padrão, mas podem ser modificados pelo usuário; são arrastáveis, móveis e destacáveis.

[3] clicar diversas vezes irá criar diversos marcadores, mas apenas o último criado ou arrastado que terá suas coordenadas lidas.

Atalhos de teclado

Principais atalhos de teclado disponíveis:

Atalho	Função
ctrl+o	Importar arquivo(s) .jpg
ctrl+d	Importar recursivamente todo o conteúdo de uma pasta
ctrl+c	Copiar metadados da entrada selecionada
ctrl+v	Colar metadados na(s) entrada(s) selecionada(s)
ctrl+s	Salvar os metadados nos campos de edição para a tabela
ctrl+shift+s	Gravar os metadados alterados nas respectivas imagens
alt+[a-z]	Focar o cursor nos menus ou campos de edição (e.g., alt+p colocará o cursor no campo **País**)
shift+e	Mostrar/esconder a doca com campos de edição
shift+g	Mostrar/esconder a doca com geolocalização
shift+t	Mostrar/esconder a doca com miniatura da imagem
shift+u	Mostrar/esconder a doca com lista de entradas modificadas
ctrl+q	Sair do programa

Algumas funções mais drásticas como limpar a tabela principal e converter a codificação dos caracteres não tem atalhos para evitar acidentes.

Tópicos

3.1 Descrição das classes e funções do VÉLIGER

Editor de metadados do banco de imagens do CEBIMar-USP.

Este programa abre imagens JPG, lê seus metadados (IPTC) e fornece uma interface para editar estas informações. Os campos foram adaptados para o propósito do banco, que é divulgar imagens com conteúdo biológico.

Campos editáveis: título, legenda, marcadores, táxon, espécie, especialista, autor, direitos, tamanho, local, cidade, estado e país.

Centro de Biologia Marinha da Universidade de São Paulo.

class **AboutDialog** (*parent*)
Janela com informações sobre o programa.

class **AutoModels** (*list*)
Cria modelos para autocompletar campos de edição.

class **CompleterLineEdit** (**args*)
Editor especial para marcadores.

Adaptado de John Schember: john.nachtimwald.com/2009/07/04/qcompleter-and-comma-separated-tags/

class **DockEditor** (*parent*)
Dock com campos para edição dos metadados.

autolistgen (*models*)
Gera autocompletadores dos campos.

savedata ( )
Salva valores dos campos para a tabela.

setcurrent (*values*)
Atualiza campos de edição quando entrada é selecionada na tabela.

setsingle (*index, value, oldvalue*)
Atualiza campo de edição correspondente quando dado é alterado.

class **DockGeo** (*parent*)
Dock para editar a geolocalização da imagem.

geodict(*string*)
Extrai coordenadas da string do editor.

get_decimal(*ref, deg, min, sec*)
Descobre o valor decimal das coordenadas.

get_exif(*filepath*)
Extrai o exif da imagem selecionada usando o pyexiv2 0.1.3.

load_geocode(*gps*)
Pega coordenadas e chama mapa com as variáveis correspondetes.

newgps()
Pega as novas coordenadas do editor.

resolve(*frac*)
Resolve a fração das coordenadas para int.

Por padrão os valores do exif são guardados como frações. Por isso é necessário converter.

setcurrent(*values*)
Mostra geolocalização da imagem selecionada.

setdms(*dms*)
Atualiza as coordenadas do editor.

state(*visible*)
Relata se aba está visível e/ou selecionada.

Captura sinal emitido pelo mainWidget.

Por não distinguir entre visível e aba selecionada não funciona em algumas situações.

un_decimal(*lat, long*)
Converte o valor decimal das coordenadas.

Retorna dicionário com referência cardinal, graus, minutos e segundos.

update_geo()
Captura as coordenadas do marcador para atualizar o editor.

write_geo()
Grava novas coordenadas nas imagens selecionadas.

write_html(*unset=0, lat=0.0, long=0.0, zoom=5*)
Carrega código HTML da QWebView com mapa do Google Maps.

Usando o API V3.

class DockThumb()
Dock para mostrar o thumbnail da imagem selecionada.

pixmapcache(*filepath*)
Cria cache para thumbnail.

resizeEvent(*event*)
Lida com redimensionamento do thumbnail.

setcurrent(*values*)
Mostra thumbnail, nome e data de modificação da imagem.

Captura sinal com valores, tenta achar imagem no cache e exibe informações.

updateThumb()
Atualiza thumbnail.

class **DockUnsaved**()
Exibe lista com imagens modificadas.

Utiliza dados do modelo em lista. Qualquer imagem modificada será adicionada à lista. Seleção na lista seleciona entrada na tabela. Gravar salva metadados de cada ítem da lista nas respectivas imagens.

clearlist()
Remove todas as entradas da lista.

delentry(*filename*)
Remove entrada da lista.

insertentry(*index, value, oldvalue*)
Insere entrada na lista.

Checa se a modificação não foi nula (valor atual == valor anterior) e se a entrada é duplicada.

matchfinder(*candidate*)
Buscador de duplicatas.

resizeEvent(*event*)
Lida com redimensionamentos.

sync_setselection(*selected, deselected*)
Sincroniza seleção da tabela com a seleção da lista.

writeselected()
Emite sinal para gravar metadados na imagem.

class **EditCompletion**(*parent*)
Editor dos valores para autocompletar campos de edição.

buildview(*modellist*)
Gera view do modelo escolhido.

insertrow()
Insere linha no modelo.

populate()
Extrai valores das fotos e popula modelo.

removerow()
Remove linha do modelo.

class **FlushFile**(*f*)
Tira o buffer do print.

Assim rola juntar prints diferentes na mesma linha. Só pra ficar bonitinho... meio inútil.

class **InitPs**()
Inicia variáveis e parâmetros globais do programa.

class **ListModel**(*parent, list, *args*)
Modelo com lista de imagens modificadas.

data(*index, role*)
Cria elementos da lista a partir dos dados.

insert_rows(*position, rows, parent, entry*)
Insere linhas.

remove_rows(*position, rows, parent*)
Remove linhas.

rowCount (*parent*)
Conta linhas.

class MainCompleter (*model, parent*)
Autocomplete principal.

class MainTable (*datalist, header, *args*)
Tabela principal com entradas.

changecurrent (*current, previous*)
Identifica a célula selecionada, extrai valores e envia para editor.

Os valores são enviados através de um sinal.

editmultiple (*index, value, oldvalue*)
Edita outras linhas selecionadas.

emitlost (*filename*)
Emite aviso para remover entrada da lista de modificados.

emitsaved ()
Emite aviso que os metadados foram gravados nos arquivos.

outputrows (*toprow*)
Identifica linhas dentro do campo de visão da tabela.

resizecols (*index*)
Ajusta largura das colunas da tabela.

update_selection (*selected, deselected*)
Pega a entrada selecionada, extrai os valores envia para editor.

Os valores são enviados através de um sinal.

class MainWindow ()
Janela principal do programa.

Inicia as instâncias dos outros componentes e aguarda interação do usuário.

cachetable ()
Salva estado atual dos dados em arquivos externos.

Cria backup dos conteúdos da tabela e da lista de imagens modificadas.

changeStatus (*status, duration=2000*)
Muda a mensagem de status da janela principal.

charconverter ()
Converte codificação de Latin-1 para UTF-8.

Pega a seleção da lista de imagens modificadas e procura a linha correspondente na tabela principal. Se o item não for encontrado o item na lista é apagado.

cleartable ()
Remove todas as entradas da tabela.

Antes de deletar checa se existem imagens não-salvas na lista.

closeEvent (*event*)
O que fazer quando o programa for fechado.

commitmeta (*entries*)
Grava os metadados modificados na imagem.

Pega lista de imagens modificadas, procura entrada na tabela principal e retorna os metadados. Chama função que gravará estes metadados na imagem. Chama função que emitirá o sinal avisando a gravação foi completada com sucesso.

copydata()
Copia metadados da entrada selecionada.

createmeta(*filepath, charset='utf-8'*)
Define as variáveis extraídas dos metadados (IPTC) da imagem.

Usa a biblioteca do arquivo iptcinfo.py e retorna lista com valores.

createthumbs(*filepath*)
Cria thumbnails para as fotos novas.

delcurrent()
Deleta a(s) entrada(s) selecionada(s) da tabela.

Verifica se a entrada a ser deletada está na lista de imagens modificadas. Se estiver, chama janela para o usuário decidir se quer apagar a entrada mesmo sem as modificações terem sido gravadas na imagem. Caso a resposta seja positiva a entrada será apagada e retirada da lista de imagens modificadas.

exiftool(*values*)
Grava os metadados no arquivo.

Usa subprocesso para chamar o exiftool, que gravará os metadados modificados na imagem.

Função não utilizada, mas pode ser útil algum dia.

imgfinder(*folder*)
Busca recursivamente imagens na pasta selecionada.

É possível definir as extensões a serem procuradas. Quando um arquivo é encontrado ele verifica se já está na tabela. Se não estiver, ele chama a função para extrair os metadados e insere uma nova entrada.

matchfinder(*candidate*)
Verifica se entrada já está na tabela.

O candidato pode ser o nome do arquivo (string) ou a entrada selecionada da tabela (lista). Retorna uma lista com duplicatas ou lista vazia caso nenhuma seja encontrada.

openabout_dialog()
Abre janela sobre o programa.

opendir_dialog()
Abre janela para selecionar uma pasta.

Chama a função para varrer a pasta selecionada.

openfile_dialog()
Abre janela para escolher arquivos.

Para selecionar arquivo(s) terminados em .jpg.

openmanual_dialog()
Abre janela do manual de instruções.

openpref_dialog()
Abre janela de opções.

pastedata()
Cola metadados na(s) entrada(s) selecionada(s).

readsettings()
Lê o estado anterior do aplicativo durante a inicialização.

setselection(*filename*)
Sincroniza seleção entre lista e tabela principal.

Pega a seleção da lista de imagens modificadas e procura a linha correspondente na tabela principal. Se o item não for encontrado o item na lista é apagado.

writemeta(*values*)
Grava os metadados no arquivo.

writesettings()
Salva estado atual do aplicativo.

class ManualDialog(*parent*)
Janela do manual de instruções.

class PrefsDialog(*parent*)
Janela de preferências.

emitrebuild()
Emite sinal com modelos atualizados e ordenados.

class PrefsGerais()
Opções gerais do programa.

class TableModel(*parent, mydata, header, *args*)
Modelo dos dados.

columnCount(*parent*)
Conta as colunas.

data(*index, role*)
Transforma dados brutos em elementos da tabela.

flags(*index*)
Indicadores do estado de cada ítem.

headerData(*col, orientation, role*)
Constrói cabeçalho da tabela.

insert_rows(*position, rows, parent, entry*)
Insere entrada na tabela.

remove_rows(*position, rows, parent*)
Remove entrada da tabela.

rowCount(*parent*)
Conta as linhas.

setData(*index, value, role*)
Salva alterações nos dados a partir da edição da tabela.

setcolor(*index, role*)
Pinta células da tabela.

sort(*col, order*)
Ordena entradas a partir de valores de determinada coluna

class TagCompleter(*model, parent*)
Completador de marcadores.

Adaptado de John Schember: john.nachtimwald.com/2009/07/04/qcompleter-and-comma-separated-tags/

CAPÍTULO 4

Índices e tabelas

- *Index*
- *Module Index*
- *Search Page*

Índice do Módulo

V

Índice

N

O

P

R

S

T

U

V

W