
Globo Photo Documentation

Versão 0.0.1

Gilson Filho

09/06/2014

1	Documentação:	3
1.1	Arquitetura	3
1.2	Instalação	4
1.3	Gerenciador de Fotos	4
1.4	Changelog	9
1.5	Roadmap	9



Globo Photo é um projeto desenvolvido para exibir as fotos que você for colocar em uma simples galeria. É exibido de forma elegante, através de um slider, que facilita na navegação.

Funcionalidades:

- Design responsivo;
- Dois tipos de gerenciamento de fotos, e com visualização de cada imagem;
- Feito com menos plugins possível, melhorando assim a performance;
- Por ser feito de forma simples, a dificuldade de evoluir é pequena.

Documentação:

1.1 Arquitetura

A Globo Photo tem sua arquitetura pensada para receber novas features como também na facilidade de melhorar funcionalidades existentes.

Ela foi desenvolvida usando as seguintes tecnologias:

- Python 2.7
- MySQL 5.6
- Django 1.6
- South 0.8.4
- Pillow 2.4.0
- dj-database-url 0.3.0
- MySQL-python 1.2.5

A estrutura do projeto está organizado da seguinte forma:

```
-- bootstrap
-- doc
-- globophoto
|  -- settings.py
|  -- templates
|  |  -- 404.html
|  |  -- 500.html
|  |  -- base.html
|  -- urls.py
|  -- wsgi.py
-- media
-- static
-- requirements.txt
-- Makefile
-- manage.py
```

- **bootstrap**: Pasta que contém a função de criar um ambiente virtual e instalar as dependências que estão no *requirements.txt*.
- **doc**: Aonde contém a documentação do projeto. Lá existem mais detalhes referentes ao mesmo.
- **globophoto**: Pasta do projeto

- **media**: Pasta que será criada quando for inserida fotos no sistema.
- **static**: Contêm os assets do projeto.
- **requirements.txt**: Armazena as dependências do projeto.
- **Makefile**: Arquivo com os comandos mais usados no projeto.
- **manage.py**: Arquivo que acessa os comandos django do projeto.

Como foi visto acima, dentro da pasta *globophoto* existem outros arquivos, que são provenientes a um projeto Django, além do *manage.py*, que são:

- **settings.py**: Arquivo que armazena as configurações gerais do projeto, como conexão com o banco de dados, módulos do sistema, etc.
- **templates**: Contêm os templates usado pelo todo o projeto. Como mostrado existe o *base.html* que possui o layout geral do sistema, o *404.html* que é exigido quando uma página não é encontrada e o *500.html* quando ocorre um erro interno.
- **urls.py**: Todas as rotas usados no sistema estão nesse arquivo.
- **wsgi.py**: Módulo responsável de preparar o projeto para ser iniciado, usando WSGI.

1.2 Instalação

Para instalar o projeto siga os procedimentos abaixo:

1. Faça o checkout do projeto:

```
$ git clone https://github.com/gilsondev/globophoto.git
```

2. Crie o ambiente virtual executando o seguinte comando:

```
$ cd globophoto
$ python bootstrap
```

3. Execute o comando abaixo para preparar o ambiente e rodar o servidor:

```
(globophoto)$ make run
```

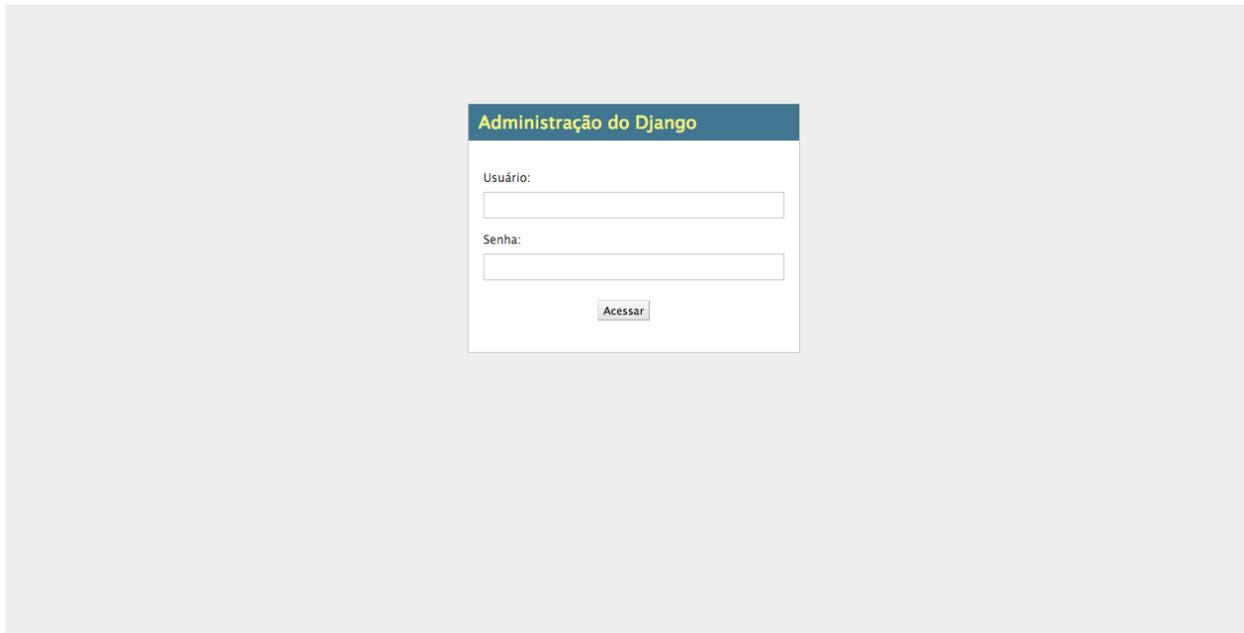
1.3 Gerenciador de Fotos

A Globo photo tem duas formas de gerenciar suas fotos: pelo django admin ou pelo gerenciador desenvolvido no projeto.

1.3.1 1. Pelo Django Admin

O Django disponibiliza uma ferramentas de administração chamado django-admin, que é preparado para esse tipo de necessidade.

Durante a configuração do projeto, ele pediu para criar um super-usuário, que é o administrador que irá acessar esse painel. Quando acessar **http://localhost:8000/admin**, vai mostrar essa página:



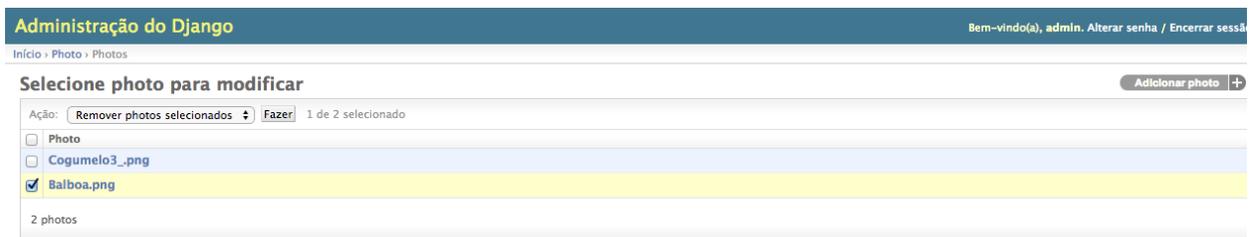
Você preenche o usuário e a senha que criou e clica em *Acessar*. Como você pode ver logo abaixo, o painel administrativo é simples e direto:



Além do gerenciador das fotos, você tem como manter usuários e papéis no sistema, caso queira. Ao entrar na seção de fotos do painel, você visualiza a lista de fotos cadastradas:



Já na própria lista de fotos, você tem a opção de editar, selecionando um dos itens da lista, como também remover. Para remover, marque a foto que deseja e selecione em *Ação* a opção *Remover photos selecionados*:



Em caso de editar, clique no nome da foto. Para criar uma foto, você vai na opção *Adicionar photo*. Clicando nesse botão, você será redirecionado para esse formulário:



Quando você anexa uma foto, você tem três opções:

- **Salvar e adicionar outro(a):** O painel vai adicionar a foto anexada, e volta para o formulário para inserir outra foto.
- **Salvar e continuar editando:** Ele salva a foto, mas volta para o formulário caso mudou de idéia, e deseja editar.
- **Salvar:** Será salvo a foto e então depois vai para a lista de fotos cadastradas.

Caso tente salvar, sem anexar uma foto, um erro será exibido:

Administração do Django Bem-vindo(a), admin. Alterar senha / Encerrar sessão

Início > Photo > Photos > Adicionar photo

Adicionar photo

Por favor, corrija os erros abaixo.

Este campo é obrigatório.

Imagem: No file chosen

1.3.2 2. Pelo Gerenciador do Globo Photo

Caso não queira ter o trabalho de logar e caso não tenha gostado do painel do Django, você tem a opção de usar o gerenciador da Globo Photo. Para acessar, clique na opção *Gerenciar as Fotos* na página da galeria.

Ao clicar, será direcionado para essa página, que irá mostrar as fotos cadastradas:

Globo Photo Galeria Gerenciar as Fotos

Listar Fotos

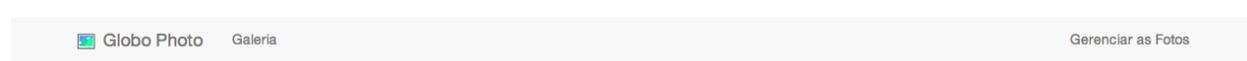
Opções	Imagem	Nome da Imagem	Link da Imagem
Editar Excluir		Balboa.png	Download da Foto
Editar Excluir		Cogumelo3_.png	Download da Foto

ou

Na lista de fotos você tem:

- **Opções:** Funções de editar ou excluir a foto selecionada.
- **Imagem:** Uma pré-visualização da foto.
- **Nome da Imagem:** É o nome da imagem anexada.
- **Download da Imagem:** Um link para você baixar a imagem selecionada.

Quando adiciona uma nova foto, clique no botão *Adicionar Foto*. O seu formulário será esse:



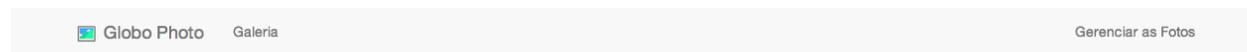
Nova Foto

Imagem:

No file chosen

ou

Anexando e salvando, será redirecionado para a lista novamente. Mas e se eu quiser editar? Simples, na lista de fotos clique na opção *Editar* da foto que deseja e terá isso:



Editar Foto



Imagem: Atualmente: [images/Balboa.png](#)

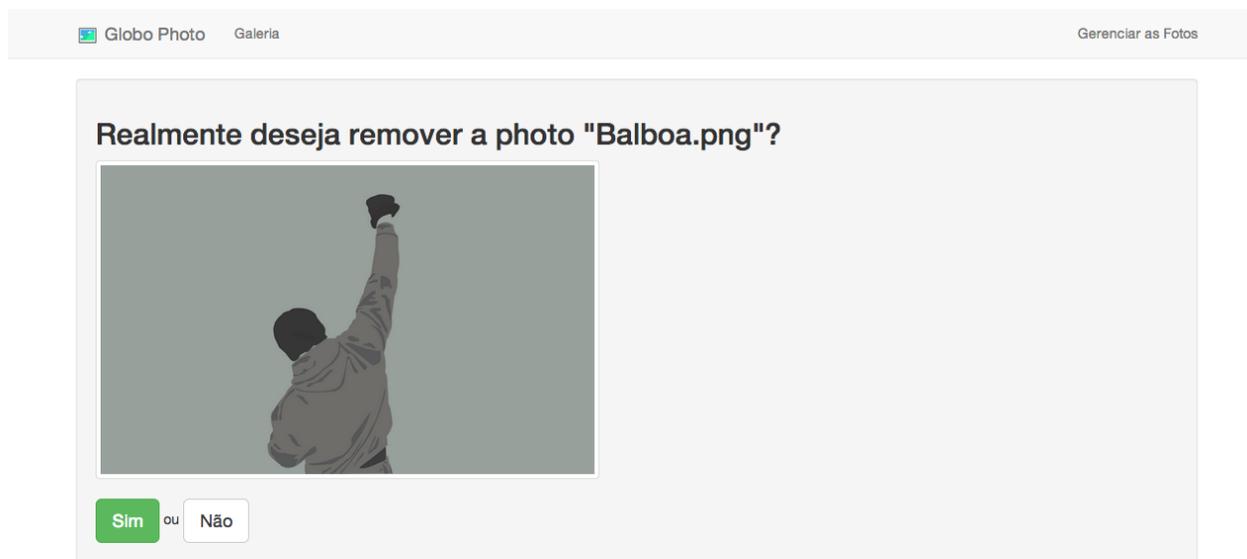
Modificar:

No file chosen

ou

Assim, você anexa outra foto e clica no *Salvar*. Feito isso, vai na lista de fotos novamente e irá na pré-visualização que a foto foi realmente substituída.

E finalmente, para excluir você clica na opção *Excluir*, e irá aparecer uma mensagem perguntando se realmente deseja fazer essa ação. Caso tenha certeza, clique em *Sim* e então a foto será removida.



1.4 Changelog

Todas as implementações de cada versão, estão listadas aqui.

1.4.1 Versão 0.0.1

- Gerenciamento de fotos pelo Django Admin;
- Gerenciamento de fotos desenvolvido sob demanda para o sistema;
- Criação da galeria de imagens, no formato de slider;
- Documentação do projeto;
- Configuração de uso para o Travis CI;

1.5 Roadmap

- Implementar API REST;
- Separar camada back-end e front-end em projetos diferentes;
- Implementar um client web, usando Backbone para consumir os serviços;
- *genindex*
- *modindex*
- *search*