

# Novidades do Django 1.2

- Suporte a múltiplos bancos de dados
- Validação de models
- Framework de mensagens
- Proteção contra CSRF
- Mudanças no contrib.auth
- Templatetag "if" mais esperta
- Mudanças no admin
- E mais ...

# Múltiplos bancos de dados ¶

- Escalabilidade
- Integração com outras aplicações
- Não é possível manter modelos relacionados (através de ForeignKey ou many-to-many) em bancos diferentes

# Múltiplos bancos de dados¶

```
DATABASES = {  
    'default': {  
        'NAME': 'app_data',  
        'ENGINE': 'django.db.backends.postgresql_psycopg2',  
        'USER': 'postgres_user',  
        'PASSWORD': 's3krit'  
    },  
    'users': {  
        'NAME': 'user_data',  
        'ENGINE': 'django.db.backends.mysql',  
        'USER': 'mysql_user',  
        'PASSWORD': 'priv4te'  
    }  
}
```

# Múltiplos bancos de dados ¶

```
QuerySet.using()
```

```
>>> User.objects.using('users').all()
```

```
Model.save(using='db')
```

```
>>> user = User(username='flavioamieiro')
```

```
>>> user.save(using='users')
```

## Database routers

Classes que definem quatro métodos:

- `db_for_read`
- `db_for_write`
- `allow_relation`
- `allow_syncdb`

# Validação de models ¶

- Validação de campos específicos
- Validação de todo o model
- Validação de unicidade de um campo
- Usado pelo ModelForm
- Não é chamado automaticamente por `Model.save()`

# Framework de mensagens¶

- Mensagens baseadas em cookies ou em sessões
- Diferentes níveis de mensagens
- É possível enviar mensagens para usuários não autenticados

# Proteção contra CSRF ¶

```
from django.views.decorators.csrf import csrf_protect
```

```
@csrf_protect
```

```
def my_view(request):
```

```
    # ...
```

```
    pass
```

```
<form action="." method="post">
```

```
    {% csrf_token %}
```

```
    <input type="text">
```

```
</form>
```

# Framework de mensagens¶

```
from django.contrib import messages

def view(request):
    messages.success(request, 'Inscreva-se na pythoncampus')
    # ...
```

```
# template.html
{% if messages %}
    <ul class="messages">
        {% for message in messages %}
            <li>{{ message }}</li>
        {% endfor %}
    </ul>
{% endif %}
```



## Mudanças no contrib.auth

- Permissões por objeto
- Permissão para usuários anônimos
- Nomes de usuários aceitam mais caracteres

# Templatetag 'if' mais esperta

```
# Django 1.1
{% ifnotequal a b %}
...
{% endifnotequal %}
```

```
# Django 1.2
{% if a != b %}
...
{% endif %}
```

operadores suportados:

```
==, !=, <, >, <=, >=, in, not in
```

# Mudanças no admin

- JQuery
- Campos somente leitura

## E mais ...¶

- Backends de e-mail
- Cache de templates
- Fixtures com 'Natural keys'
- Fail Fast para testes
- BigIntegerField
- Melhorias na localização
- Realce de sintaxe nos comandos do django-admin
- Feeds podem ser usados diretos como views

- <http://docs.djangoproject.com/en/dev/releases/1.2/>
- <http://djangoadvent.com/>
- <http://www.github.com/django/django>