

Generated Documentation



Contents

Package app-Model Classes	1
Class AppModel	1
Package Controller Procedural Elements	3
AppModel.php	3
Package Controller Classes	4
Class Usuario	4
Var \$hasMany	5
Var \$name	5
Var \$validate	6
Method beforeSave	6
Method checa_dir	6
Method checa_nome	7
Method move_arquivos	7
Method passwordConfirmation	8
Method trata_nome	8
Method upload	8
Package Model Procedural Elements	11
Category.php	11
Package Model Classes	12
Class Category	12
Var \$hasMany	12
Var \$validate	13
Comentario.php	14
Class Comentario	14
Var \$belongsTo	15
Var \$validate	15
Mensagem.php	17
Class Mensagem	17
Var \$validate	18
Negociacao.php	19
Produto.php	20
Report.php	21
Class Negociacao	21
Var \$belongsTo	22
Class Produto	22
Var \$belongsTo	23
Var \$hasMany	23
Var \$validate	24
Method beforeSave	24
Method checa_dir	25
Method checa_nome	25
Method move_arquivos	25

Method trata_nome	26
Method upload	26
Class Report	27
Var \$belongsTo	27
Var \$validate	28
Servico.php	29
Class Service	29
Var \$belongsTo	30
Var \$hasMany	30
Var \$validate	31
Method beforeSave	31
Method checa_dir	32
Method checa_nome	32
Method move_arquivos	33
Method trata_nome	33
Method upload	33
Appendices	35
Appendix A - Class Trees	36
Controller	36
app-Model	36
Model	36

Package app-Model Classes

Class AppModel

[line 54]

Application model for Cake.

Add your application-wide methods in the class below, your models will inherit them.

- **Package** app-Model

Package Controller Procedural Elements

AppModel.php

- **Package** Controller
- **Sub-Package** AppModel
- **Author** Fabio Santos < fabiosantos1388@gmail.com>
- **Author** Higor Vital < h.vital196@gmail.com>
- **Author** Dermeval Neves < dermevalneves@gmail.com>
- **Author** Sillas Senna < ssennarm@gmail.com>
- **Author** Adriano Cavalcante < nuno.cavalcante@gmail.com>
- **Author** J  f  ssica Santos jessica22san@gmail.com>
- **Author** Arthur Hagn  s < arthurecomp@gmail.com>
- **Author** Nielson V  jgno < nielsonvagno@gmail.com>
- **Author** Marcelo Bi  o < marcelobiao2@gmail.com>
- **Author** Leno Oliveira < lenoosouza@gmail.com>
- **Author** Joacy Mesquita < joacymsilva@gmail.com>
- **Author** Kaique Cabral < kaaiquecabral@gmail.com>
- **Version** 1.0
- **Copyright** 2014, SoftComp Empreendimentos, 2005-2011, Cake Software Foundation, Inc. (<http://cakefoundation.org>)
- **Copyright** Copyright (c) Cake Software Foundation, Inc. (<http://cakefoundation.org>)
- **Link** [CakePHP\(tm\) Project](#)
- **Link** [CakePHP\(tm\) Project](#)
- **Since** CakePHP(tm) v 0.2.9
- **Since** CakePHP(tm) v 0.2.9
- **License** MIT
- **License** [MIT License](#)
- **Example** example not found

Package Controller Classes

Class Usuario

[line 27]

- **Package** Controller
- **Sub-Package** UsuariosController
- **Author** Sillas Senna < ssennarm@gmail.com>
- **Author** Higor Vital < h.vital96@gmail.com>
- **Author** Dermeval Neves < dermevalneves@gmail.com>
- **Author** Arthur HagnÃas < arthurecomp@gmail.com>
- **Author** Adriano Cavalcante < nuno.cavalcante@gmail.com>
- **Author** JÃfÃssica Santos jessica22san@gmail.com>
- **Author** Fabio Santos < fabiosantos1388@gmail.com>
- **Author** Nielson VÃjgno < nielsonvagno@gmail.com>
- **Author** Marcelo BiÃ£o < marcelobiao2@gmail.com>
- **Author** Leno Oliveira < lenoosouza@gmail.com>
- **Author** Kaique Cabral < kaaiquecabral@gmail.com>
- **Author** Joacy Mesquita < joacymsilva@gmail.com>
- **Version** 1.0
- **Copyright** 2014, SoftComp Empreendimentos, 2005-2011, Cake Software Foundation, Inc. (<http://cakefoundation.org>)
- **Link** [CakePHP\(tm\) Project](#)
- **Since** CakePHP(tm) v 0.2.9
- **License** MIT
- **Example** example not found

Usuario::\$hasMany

```
array = array(
    'Comentario' => array(
        'className' => 'Comentario',
        'foreignKey' => 'usuario_id',
        'dependent' => false,
        'conditions' => "",
        'fields' => "",
        'order' => "",
        'limit' => "",
```

```
'offset' => ",
'exclusive' => ",
'finderQuery' => ",
'counterQuery' => "
```

```
), 'Negociacao' => array('className' => 'Negociacao', 'foreignKey' => 'usuario_id', 'dependent' => false, 'conditions' => "", 'fields' => "", 'order' => "", 'limit' => "", 'offset' => "", 'exclusive' => "", 'finderQuery' => "", 'counterQuery' => ""), 'Produto' => array('className' => 'Produto', 'foreignKey' => 'usuario_id', 'dependent' => false, 'conditions' => "", 'fields' => "", 'order' => "", 'limit' => "", 'offset' => "", 'exclusive' => "", 'finderQuery' => "", 'counterQuery' => ""), 'Servico' => array('className' => 'Servico', 'foreignKey' => 'usuario_id', 'dependent' => false, 'conditions' => "", 'fields' => "", 'order' => "", 'limit' => "", 'offset' => "", 'exclusive' => "", 'finderQuery' => "", 'counterQuery' => "")) [line 253]
```

hasMany associations

- **Access public**

Usuario::\$name

```
mixed = 'Usuario' [line 29]
```

- **Access public**

Usuario::\$validate

```
array = array(
    'Email' => array(
        'notEmpty' => array(
            'rule' => array('email'), 'message' => 'Você deve colocar o Email corretamente.', '//allowEmpty' => false,
            '//required' => false,
            '//last' => false, // Stop validation after this rule
            '//on' => 'create', // Limit validation to 'create' or 'update' operations
        ), 'name' => array('notEmpty' => array('rule' => array('notEmpty'), 'message' => 'Digite seu nome.', '//allowEmpty' => false,
            '//required' => false,
            '//last' => false, // Stop validation after this rule
            '//on' => 'create', // Limit validation to 'create' or 'update' operations
        ), 'Senha' => array('notEmpty' => array('rule' => array('notEmpty'), 'message' => 'Coloque uma senha confiável', '//allowEmpty' => false,
            '//required' => false,
            '//last' => false, // Stop validation after this rule
            '//on' => 'create', // Limit validation to 'create' or 'update' operations
        ), 'minLength' => array('rule' => array('minLength', 6), 'required' => true, 'message' => 'Sua senha precisa conter pelo menos 6 caracteres'), 'Senha_Confirmacao' => array('notEmpty' => array('rule' => 'notEmpty', 'required' => true, 'message' =>
```



```

>'Confirme sua senha'),'minLength'=>array('rule'=>array('minLength',6),'required'=>true,'message'=>'Sua
senha precisa conter pelo menos 6
caracteres'),'passwordConfirmation'=>array('rule'=>array('passwordConfirmation'),'message'=>'As duas
senhas n o conferem'),), 'Telefone1'=>array('notEmpty'=>array('rule'=>'notEmpty','message'=>'Coloque seu
telefone','allowEmpty' => false,
//required' => false,
//last' => false, // Stop validation after this rule
//on' => 'create', // Limit validation to 'create' or 'update' operations
),), 'Sexo'=>array('notEmpty'=>array('rule'=>array('notEmpty'),'message' => 'Your custom message here',
//allowEmpty' => false,
//required' => false,
//last' => false, // Stop validation after this rule
//on' => 'create', // Limit validation to 'create' or 'update' operations
),),) [line 150]

```

Validation rules

- **Access public**

void function Usuario::beforeSave([\$options = array()]) [line 38]

Function Parameters:

- **\$options**

beforeSave Esse m todo   executado antes de salvar as informa  es de usu rio.

Nele,   testado se o usu rio enviou uma foto, caso sim, a foto   upada para o servidor, caso n o, a foto   setada para null. Ap s isso,   adicionado ao AuthComponent as informa  es de usu rio e senha.

- **Access public**

void function Usuario::checa_dir(\$dir) [line 84]

Function Parameters:

- *string* **\$dir**

MÃ©todo `checa_dir` Verifica se o diretÃ³rio existe, se nÃ£o ele cria.

- **Access public**

string function Usuario::checa_nome(\$imagem, \$dir, \$data) [line 101]

Function Parameters:

- *Array* **\$imagem**
- *String* **\$data**
- **\$dir**

MÃ©todo `checa_nome`

Verifica se o nome do arquivo jÃ¡ existe, se existir adiciona um numero ao nome e verifica novamente

- **Access public**

void function Usuario::move_arquivos(\$imagem, \$dir, \$data) [line 136]

Function Parameters:

- *Array* **\$imagem**
- *String* **\$data**
- **\$dir**

MÃ©todo `move_arquivos` Move o arquivo para a pasta de destino.

- **Access public**

void function Usuario::passwordConfirmation(\$data) [*line 231*]

Function Parameters:

- **\$data**

- **Access public**

void function Usuario::trata_nome(\$imagem_nome) [*line 123*]

Function Parameters:

- *String* **\$imagem_nome**

Método trata_nome Trata o nome removendo espaços, acentos e caracteres em maiúsculo.

- **Access public**

void function Usuario::upload([\$imagem = array()], [\$dir = 'img']) [*line 60*]

Function Parameters:

- *array* **\$imagem** Array com informações da imagem
- *string* **\$dir** Diretório da imagem.

upload Método para fazer upload de uma imagem no banco de dados.

- **Access public**

Package Model Procedural Elements

Category.php

- **Package** Model
- **Sub-Package** Category
- **Author** Sillas Senna < ssennarm@gmail.com>
- **Author** Higor Vital < h.vital96@gmail.com>
- **Author** Dermeval Neves < dermevalneves@gmail.com>
- **Author** Arthur HagnÃas < arthurecomp@gmail.com>
- **Author** Adriano Cavalcante < nuno.cavalcante@gmail.com>
- **Author** JÃfÃssica Santos jessica22san@gmail.com>
- **Author** Fabio Santos < fabiosantos1388@gmail.com>
- **Author** Nielson VÃjgno < nielsonvagno@gmail.com>
- **Author** Marcelo BiÃ£o < marcelobiao2@gmail.com>
- **Author** Leno Oliveira < lenoosouza@gmail.com>
- **Author** Kaique Cabral < kaaiquecabral@gmail.com>
- **Author** Joacy Mesquita < joacymsilva@gmail.com>
- **Version** 1.0
- **Copyright** 2014, SoftComp Empreendimentos, 2005-2011, Cake Software Foundation, Inc. (<http://cakefoundation.org>)
- **Link** [CakePHP\(tm\) Project](#)
- **Since** CakePHP(tm) v 0.2.9
- **License** MIT
- **Example** example not found

Package Model Classes

Class Category

[line 32]

Category Model

- **Package** Model
- **Sub-Package** Category
- **Property** [Produto](#) \$Produto:
- **Property** [Servico](#) \$Servico:
- **Property** Terceiro \$Terceiro:

Category::\$hasMany

```
array = array(
    'Produto' => array(
        'className' => 'Produto',
        'foreignKey' => 'category_id',
        'dependent' => false,
        'conditions' => "",
        'fields' => "",
        'order' => "",
        'limit' => "",
        'offset' => "",
        'exclusive' => "",
        'finderQuery' => "",
        'counterQuery' => ""
    ),
    'Servico' => array('className' => 'Servico', 'foreignKey' => 'category_id', 'dependent' => false, 'conditions' => "", 'fields' => "", 'order' => "", 'limit' => "", 'offset' => "", 'exclusive' => "", 'finderQuery' => "", 'counterQuery' => ""),
    'Terceiro' => array('className' => 'Terceiro', 'foreignKey' => 'category_id', 'dependent' => false, 'conditions' => "", 'fields' => "", 'order' => "", 'limit' => "", 'offset' => "", 'exclusive' => "", 'finderQuery' => "", 'counterQuery' => "")) [line 59]
```

hasMany associations

- **Access** public

Category::\$validate

```
array = array(
    'name' => array(
        'notEmpty' => array(
            'rule' => array('notEmpty'), // 'message' => 'Your custom message here',
            // 'allowEmpty' => false,
            // 'required' => false,
            // 'last' => false, // Stop validation after this rule
            // 'on' => 'create', // Limit validation to 'create' or 'update' operations
        ), ), [line 39]
```

Regras de validação

- **Access** public

Comentario.php

- **Package** Model
- **Sub-Package** Comentario
- **Author** Sillas Senna < ssennarm@gmail.com>
- **Author** Higor Vital < h.vital96@gmail.com>
- **Author** Dermeval Neves < dermevalneves@gmail.com>
- **Author** Arthur HagnÃas < arthurecomp@gmail.com>
- **Author** Adriano Cavalcante < nuno.cavalcante@gmail.com>
- **Author** JÃfÃssica Santos jessica22san@gmail.com>
- **Author** Fabio Santos < fabiosantos1388@gmail.com>
- **Author** Nielson VÃjgno < nielsonvagno@gmail.com>
- **Author** Marcelo BiÃ£o < marcelobiao2@gmail.com>
- **Author** Leno Oliveira < lenoosouza@gmail.com>
- **Author** Kaique Cabral < kaaiquecabral@gmail.com>
- **Author** Joacy Mesquita < joacymsilva@gmail.com>
- **Version** 1.0
- **Copyright** 2014, SoftComp Empreendimentos, 2005-2011, Cake Software Foundation, Inc. (<http://cakefoundation.org>)
- **Link** [CakePHP\(tm\) Project](#)
- **Since** CakePHP(tm) v 0.2.9
- **License** MIT
- **Example** example not found

Class Comentario

[line 32]

Comentario Model

- **Package** Model
- **Sub-Package** Comentario
- **Property** [Usuario](#) \$Usuario:
- **Property** [Servico](#) \$Servico:
- **Property** [Produto](#) \$Produto:

Comentario::\$belongsTo

```
array = array(  
    'Usuario' => array(  

```

```

        'className' => 'Usuario',
        'foreignKey' => 'usuario_id',
        'conditions' => "",
        'fields' => "",
        'order' => ""

    ), 'Servico' => array('className' => 'Servico', 'foreignKey' => 'servico_id', 'conditions' => "", 'fields' => "", 'order' => ""), 'Produto' => array('className' => 'Produto', 'foreignKey' => 'produto_id', 'conditions' => "", 'fields' => "", 'order' => "")) [line 91]

```

Associação belongsTo

- **Access public**

Comentário::\$validate

```

array = array(
    'Data' => array(
        'date' => array(
            'rule' => array('date'), //message' => 'Your custom message here',
            //allowEmpty' => false,
            //required' => false,
            //last' => false, // Stop validation after this rule
            //on' => 'create', // Limit validation to 'create' or 'update' operations
        ), 'Texto' => array('notEmpty' => array('rule' => array('notEmpty'), //message' => 'Your custom message here',
            //allowEmpty' => false,
            //required' => false,
            //last' => false, // Stop validation after this rule
            //on' => 'create', // Limit validation to 'create' or 'update' operations
        ), 'minLength' => array('rule' => array('minLength', 20), 'required' => true, 'message' => 'Mínimo de 20 caracteres'),
        'maxLength' => array('rule' => array('maxLength', 150), 'required' => true, 'message' => 'Máximo de 150 caracteres'),
        'usuario_id' => array('numeric' => array('rule' => array('numeric'), //message' => 'Your custom message here',
            //allowEmpty' => false,
            //required' => false,
            //last' => false, // Stop validation after this rule
            //on' => 'create', // Limit validation to 'create' or 'update' operations
        ),) [line 39]

```

Regras de Validação

As associações abaixo foram criadas com todas as chaves possíveis, as quais não forem necessárias poderão ser removidas

- **Access public**

Mensagem.php

- **Package** Model
- **Sub-Package** Mensagem
- **Author** Sillas Senna < ssennarm@gmail.com>
- **Author** Higor Vital < h.vital96@gmail.com>
- **Author** Dermeval Neves < dermevalneves@gmail.com>
- **Author** Arthur HagnÃas < arthurecomp@gmail.com>
- **Author** Adriano Cavalcante < nuno.cavalcante@gmail.com>
- **Author** JÃfÃssica Santos jessica22san@gmail.com>
- **Author** Fabio Santos < fabiosantos1388@gmail.com>
- **Author** Nielson VÃjgno < nielsonvagno@gmail.com>
- **Author** Marcelo BiÃ£o < marcelobiao2@gmail.com>
- **Author** Leno Oliveira < lenoosouza@gmail.com>
- **Author** Kaique Cabral < kaaiquecabral@gmail.com>
- **Author** Joacy Mesquita < joacymsilva@gmail.com>
- **Version** 1.0
- **Copyright** 2014, SoftComp Empreendimentos, 2005-2011, Cake Software Foundation, Inc. (<http://cakefoundation.org>)
- **Link** [CakePHP\(tm\) Project](#)
- **Since** CakePHP(tm) v 0.2.9
- **License** MIT
- **Example** example not found

Class Mensagem

[line 30]

Mensagem Model

- **Package** Model
- **Sub-Package** Mensagem
- **Access** public

Mensagem::\$validate

```
array = array(
    'Assunto' => array(
        'notEmpty' => array(
            'rule' => array('notEmpty'), // 'message' => 'Your custom message here',
```

```
//'allowEmpty' => false,
//'required' => false,
//'last' => false, // Stop validation after this rule
//'on' => 'create', // Limit validation to 'create' or 'update' operations
), 'minLength' => array('rule' => array('minLength', 10), 'required' => true, 'message' => 'Nome muito
curto'), 'maxLength' => array('rule' => array('maxLength', 30), 'required' => true, 'message' => 'Nome muito
longo'), 'Texto' => array('notEmpty' => array('rule' => array('notEmpty'), // 'message' => 'Your custom message
here',
//'allowEmpty' => false,
//'required' => false,
//'last' => false, // Stop validation after this rule
//'on' => 'create', // Limit validation to 'create' or 'update' operations
), 'minLength' => array('rule' => array('minLength', 20), 'required' => true, 'message' => 'MÃ-nimo de 20
caracteres'), 'maxLength' => array('rule' => array('maxLength', 150), 'required' => true, 'message' => 'MÃximo de
150 caracteres'),),) [line 37]
```

Regras de validaÃ§Ã£o

- **Access public**

Negociacao.php

- **Package** Model
- **Sub-Package** Negociacao
- **Author** Sillas Senna < ssennarm@gmail.com>
- **Author** Higor Vital < h.vital96@gmail.com>
- **Author** Dermeval Neves < dermevalneves@gmail.com>
- **Author** Arthur HagnÃas < arthurecomp@gmail.com>
- **Author** Adriano Cavalcante < nuno.cavalcante@gmail.com>
- **Author** JÃfÃssica Santos jessica22san@gmail.com>
- **Author** Fabio Santos < fabiosantos1388@gmail.com>
- **Author** Nielson VÃjgno < nielsonvagno@gmail.com>
- **Author** Marcelo BiÃ£o < marcelobiao2@gmail.com>
- **Author** Leno Oliveira < lenoosouza@gmail.com>
- **Author** Kaique Cabral < kaaiquecabral@gmail.com>
- **Author** Joacy Mesquita < joacymsilva@gmail.com>
- **Version** 1.0
- **Copyright** 2014, SoftComp Empreendimentos, 2005-2011, Cake Software Foundation, Inc. (<http://cakefoundation.org>)
- **Link** [CakePHP\(tm\) Project](#)
- **Since** CakePHP(tm) v 0.2.9
- **License** MIT
- **Example** example not found

Produto.php

- **Package** Model
- **Sub-Package** Negociacao
- **Author** Sillas Senna < ssennarm@gmail.com>
- **Author** Higor Vital < h.vital96@gmail.com>
- **Author** Dermeval Neves < dermevalneves@gmail.com>
- **Author** Arthur HagnÃas < arthurecomp@gmail.com>
- **Author** Adriano Cavalcante < nuno.cavalcante@gmail.com>
- **Author** JÃfÃssica Santos jessica22san@gmail.com>
- **Author** Fabio Santos < fabiosantos1388@gmail.com>
- **Author** Nielson VÃjgno < nielsonvagno@gmail.com>
- **Author** Marcelo BiÃ£o < marcelobiao2@gmail.com>
- **Author** Leno Oliveira < lenoosouza@gmail.com>
- **Author** Kaique Cabral < kaaiquecabral@gmail.com>
- **Author** Joacy Mesquita < joacymsilva@gmail.com>
- **Version** 1.0
- **Copyright** 2014, SoftComp Empreendimentos, 2005-2011, Cake Software Foundation, Inc. (<http://cakefoundation.org>)
- **Link** [CakePHP\(tm\) Project](#)
- **Since** CakePHP(tm) v 0.2.9
- **License** MIT
- **Example** example not found

Report.php

- **Package** Model
- **Sub-Package** Negociacao
- **Author** Sillas Senna < ssennarm@gmail.com>
- **Author** Higor Vital < h.vital96@gmail.com>
- **Author** Dermeval Neves < dermevalneves@gmail.com>
- **Author** Arthur HagnÃs < arthurecomp@gmail.com>
- **Author** Adriano Cavalcante < nuno.cavalcante@gmail.com>
- **Author** JÃfÃssica Santos jessica22san@gmail.com>
- **Author** Fabio Santos < fabiosantos1388@gmail.com>
- **Author** Nielson VÃjgno < nielsonvagno@gmail.com>
- **Author** Marcelo BiÃ£o < marcelobiao2@gmail.com>
- **Author** Leno Oliveira < lenoosouza@gmail.com>
- **Author** Kaique Cabral < kaaiquecabral@gmail.com>
- **Author** Joacy Mesquita < joacymsilva@gmail.com>
- **Version** 1.0
- **Copyright** 2014, SoftComp Empreendimentos, 2005-2011, Cake Software Foundation, Inc. (<http://cakefoundation.org>)
- **Link** [CakePHP\(tm\) Project](#)
- **Since** CakePHP(tm) v 0.2.9
- **License** MIT
- **Example** example not found

Class Negociacao

[line 34]

Negociacao Model As associaÃ§Ãµes abaixo foram criadas com todas as chaves possÃ-veis, as quais nÃ£o forem necessÃrias poderam ser removidas

- **Package** Model
- **Sub-Package** Negociacao
- **Property** [Usuario](#) \$Usuario:
- **Property** [Produto](#) \$Produto:
- **Property** [Servico](#) \$Servico:

Negociacao::\$belongsTo


```

array = array(
    'Usuario' => array(
        'className' => 'Usuario',
        'foreignKey' => 'usuario_id',
        'conditions' => "",
        'fields' => "",
        'order' => ""
    ),
    'Produto' => array('className' => 'Produto', 'foreignKey' => 'produto_id', 'conditions' => "", 'fields' => "", 'order' => ""),
    'Servico' => array('className' => 'Servico', 'foreignKey' => 'servico_id', 'conditions' => "", 'fields' => "", 'order' => "")
)

```

Associação belongsTo

- **Access** public

Class Produto

[line 34]

Produto Model

- **Package** Model
- **Sub-Package** Negociacao
- **Property** [Usuario](#) \$Usuario:
- **Property** [Category](#) \$Category:
- **Property** [Comentario](#) \$Comentario:
- **Property** [Negociacao](#) \$Negociacao:
- **Property** [Report](#) \$Report:

Produto::\$belongsTo

```

array = array(
    'Usuario' => array(
        'className' => 'Usuario',
        'foreignKey' => 'usuario_id',
        'conditions' => "",
        'fields' => "",
        'order' => ""
    )
)

```

```
), 'Category' => array('className' => 'Category', 'foreignKey' => 'category_id', 'conditions' => '', 'fields' => '', 'order' => ''
)) [line 142]
```

Associação belongsTo

- Access public

Produto::\$hasMany

```
array = array(
    'Comentario' => array(
        'className' => 'Comentario',
        'foreignKey' => 'produto_id',
        'dependent' => false,
        'conditions' => '',
        'fields' => '',
        'order' => '',
        'limit' => '',
        'offset' => '',
        'exclusive' => '',
        'finderQuery' => '',
        'counterQuery' => ''
    ), 'Negociacao' => array('className' => 'Negociacao', 'foreignKey' => 'produto_id', 'dependent' => false, 'conditions'
=> '', 'fields' => '', 'order' => '', 'limit' => '', 'offset' => '', 'exclusive' => '', 'finderQuery' => '', 'counterQuery' => ''), 'Report' => arr
ay('className' => 'Report', 'foreignKey' => 'produto_id', 'dependent' => false, 'conditions' => '', 'fields' => '', 'order' => '', '
limit' => '', 'offset' => '', 'exclusive' => '', 'finderQuery' => '', 'counterQuery' => '')) [line 164]
```

Associação hasMany

- Access public

Produto::\$validate

```
array = array(
    'name' => array(
        'notEmpty' => array(
            'rule' => array('notEmpty'), // 'message' => 'Your custom message here',
            // 'allowEmpty' => false,
            // 'required' => false,
            // 'last' => false, // Stop validation after this rule
        )
    )
)
```

```
//on' => 'create', // Limit validation to 'create' or 'update' operations
), 'minLength'=>array('rule'=>array('minLength',3), 'required'=>true, 'message'=>'Nome muito
curto'), 'maxLength'=>array('rule'=>array('maxLength',30), 'required'=>true, 'message'=>'Nome muito
longo'), 'Descricao'=>array('notEmpty'=>array('rule'=>array('notEmpty'), // 'message' => 'Your custom
message here',
//allowEmpty' => false,
//required' => false,
//last' => false, // Stop validation after this rule
//on' => 'create', // Limit validation to 'create' or 'update' operations
), 'minLength'=>array('rule'=>array('minLength',20), 'required'=>true, 'message'=>'MÃ-nimo de 20
caracteres'), 'maxLength'=>array('rule'=>array('maxLength',150), 'required'=>true, 'message'=>'MÃximo de
150 caracteres'), 'Preco'=>array('numeric'=>array('rule'=>array('decimal',2), // 'message' => 'Your custom
message here',
//allowEmpty' => false,
//required' => false,
//last' => false, // Stop validation after this rule
//on' => 'create', // Limit validation to 'create' or 'update' operations
), 'Quantidade'=>array('numeric'=>array('rule'=>array('numeric'), // 'message' => 'Your custom message here',
//allowEmpty' => false,
//required' => false,
//last' => false, // Stop validation after this rule
//on' => 'create', // Limit validation to 'create' or 'update' operations
), 'usuario_id'=>array('numeric'=>array('rule'=>array('numeric'), // 'message' => 'Your custom message here',
//allowEmpty' => false,
//required' => false,
//last' => false, // Stop validation after this rule
//on' => 'create', // Limit validation to 'create' or 'update' operations
), 'category_id'=>array('numeric'=>array('rule'=>array('numeric'), // 'message' => 'Your custom message here',
//allowEmpty' => false,
//required' => false,
//last' => false, // Stop validation after this rule
//on' => 'create', // Limit validation to 'create' or 'update' operations
),) [line 42]
```

Regras de validação As associações abaixo foram criadas com todas as chaves possíveis, as quais não forem necessárias poderão ser removidas

- Access public

`void function Produto::beforeSave([$options = array()]) [line 214]`

Function Parameters:

- \$options

Método beforeSave

- **Access public**

void function Produto::checa_dir(\$dir, \$imagem, \$data) [line 253]

Function Parameters:

- *Array* **\$imagem**
- *String* **\$data**
- **\$dir**

Método checa_dir Verifica se o diretório existe, se não ele cria.

- **Access public**

nome function Produto::checa_nome(\$imagem, \$dir, \$data) [line 270]

Function Parameters:

- *Array* **\$imagem**
- *String* **\$data**
- **\$dir**

Método checa_nome

Verifica se o nome do arquivo já existe, se existir adiciona um numero ao nome e verifica novamente

- **Access public**

void function Produto::move_arquivos(\$imagem, \$dir, \$data) [line 306]

Function Parameters:

- *Array* **\$imagem**
- *String* **\$data**
- **\$dir**

MÃ©todo move_arquivos Move o arquivo para a pasta de destino.

- **Access public**

void function Produto::trata_nome(\$imagem_nome, \$imagem, \$data) [line 293]

Function Parameters:

- *Array* **\$imagem**
- *String* **\$data**
- **\$imagem_nome**

MÃ©todo trata_nome Trata o nome removendo espaÃ§os, acentos e caracteres em maiÃºsculo.

- **Access public**

void function Produto::upload([\$imagem = array()], [\$dir = 'img']) [line 228]

Function Parameters:

- **\$imagem**
- **\$dir**

MÃ©todo upload

Esta funÃ§Ã£o permite ao usuÃ¡rio fazer o upload da imagem para um produto a ser cadastrado

- **Access** public

Class Report

[line 32]

Report Model

- **Package** Model
- **Sub-Package** Negociacao
- **Property** [Produto](#) \$Produto:
- **Property** [Servico](#) \$Servico:
- **Property** [Usuario](#) \$Usuario:

Report::\$belongsTo

```
array = array(
    'Produto' => array(
        'className' => 'Produto',
        'foreignKey' => 'produto_id',
        'conditions' => "",
        'fields' => "",
        'order' => ""
    ),
    'Servico' => array('className' => 'Servico', 'foreignKey' => 'servico_id', 'conditions' => "", 'fields' => "", 'order' => ""),
    'Usuario' => array('className' => 'Usuario', 'foreignKey' => 'usuario_id', 'conditions' => "", 'fields' => "", 'order' => "") [line 89]
```

Associations belongsTo

- **Access** public

Report::\$validate

```
array = array(
    'Texto' => array(
        'notEmpty' => array(
            'rule' => array('notEmpty'), // 'message' => 'Your custom message here',
            // 'allowEmpty' => false,
            // 'required' => false,
            // 'last' => false, // Stop validation after this rule
            // 'on' => 'create', // Limit validation to 'create' or 'update' operations
        ), 'minLength' => array('rule' => array('minLength', 20), 'required' => true, 'message' => 'Mínimo de 20 caracteres'), 'maxLength' => array('rule' => array('maxLength', 150), 'required' => true, 'message' => 'Máximo de 150 caracteres'), 'Data' => array('date' => array('rule' => array('date'), // 'message' => 'Your custom message here',
            // 'allowEmpty' => false,
            // 'required' => false,
            // 'last' => false, // Stop validation after this rule
            // 'on' => 'create', // Limit validation to 'create' or 'update' operations
        ), 'usuario_id' => array('numeric' => array('rule' => array('numeric'), // 'message' => 'Your custom message here',
            // 'allowEmpty' => false,
            // 'required' => false,
            // 'last' => false, // Stop validation after this rule
            // 'on' => 'create', // Limit validation to 'create' or 'update' operations
        ),) [line 39]
```

Regras de Validação

As associações abaixo foram criadas com todas as chaves possíveis, as quais não forem necessárias poderão ser removidas

- **Access public**

Servico.php

- **Package** Model
- **Sub-Package** Servico
- **Author** Sillas Senna < ssennarm@gmail.com>
- **Author** Higor Vital < h.vital96@gmail.com>
- **Author** Dermeval Neves < dermevalneves@gmail.com>
- **Author** Arthur HagnÃas < arthurecomp@gmail.com>
- **Author** Adriano Cavalcante < nuno.cavalcante@gmail.com>
- **Author** JÃfÃssica Santos jessica22san@gmail.com>
- **Author** Fabio Santos < fabiosantos1388@gmail.com>
- **Author** Nielson VÃjgno < nielsonvagno@gmail.com>
- **Author** Marcelo BiÃ£o < marcelobiao2@gmail.com>
- **Author** Leno Oliveira < lenoosouza@gmail.com>
- **Author** Kaique Cabral < kaaiquecabral@gmail.com>
- **Author** Joacy Mesquita < joacymsilva@gmail.com>
- **Version** 1.0
- **Copyright** 2014, SoftComp Empreendimentos, 2005-2011, Cake Software Foundation, Inc. (<http://cakefoundation.org>)
- **Link** [CakePHP\(tm\) Project](#)
- **Since** CakePHP(tm) v 0.2.9
- **License** MIT
- **Example** example not found

Class Servico

[line 34]

Servico Model

- **Package** Model
- **Sub-Package** Servico
- **Property** [Usuario](#) \$Usuario:
- **Property** [Category](#) \$Category:
- **Property** [Comentario](#) \$Comentario:
- **Property** [Negociacao](#) \$Negociacao:
- **Property** [Report](#) \$Report:

Servico::\$belongsTo


```

array = array(
    'Usuario' => array(
        'className' => 'Usuario',
        'foreignKey' => 'usuario_id',
        'conditions' => "",
        'fields' => "",
        'order' => ""
    ), 'Category' => array('className' => 'Category', 'foreignKey' => 'category_id', 'conditions' => "", 'fields' => "", 'order' => ""
)) [line 128]

```

belongsTo associations

- **Access public**

Service::\$hasMany

```

array = array(
    'Comentario' => array(
        'className' => 'Comentario',
        'foreignKey' => 'servico_id',
        'dependent' => false,
        'conditions' => "",
        'fields' => "",
        'order' => "",
        'limit' => "",
        'offset' => "",
        'exclusive' => "",
        'finderQuery' => "",
        'counterQuery' => ""
    ), 'Negociacao' => array('className' => 'Negociacao', 'foreignKey' => 'servico_id', 'dependent' => false, 'conditions' => "", 'fields' => "", 'order' => "", 'limit' => "", 'offset' => "", 'exclusive' => "", 'finderQuery' => "", 'counterQuery' => ""), 'Report' => array('className' => 'Report', 'foreignKey' => 'servico_id', 'dependent' => false, 'conditions' => "", 'fields' => "", 'order' => "", 'limit' => "", 'offset' => "", 'exclusive' => "", 'finderQuery' => "", 'counterQuery' => "") [line 150]

```

AssociaÃ§Ãµes hasMany

- **Access public**

Service::\$validate

```

array = array(

```

```

        'name' => array(
            'notEmpty' => array(
                'rule' => array('notEmpty'), // 'message' => 'Your custom message here',
                // 'allowEmpty' => false,
                // 'required' => false,
                // 'last' => false, // Stop validation after this rule
                // 'on' => 'create', // Limit validation to 'create' or 'update' operations
            ), 'minLength' => array('rule' => array('minLength', 10), 'required' => true, 'message' => 'Nome muito
curto'), 'maxLength' => array('rule' => array('maxLength', 30), 'required' => true, 'message' => 'Nome muito
longo'), 'Descricao' => array('notEmpty' => array('rule' => array('notEmpty'), // 'message' => 'Your custom
message here',
                // 'allowEmpty' => false,
                // 'required' => false,
                // 'last' => false, // Stop validation after this rule
                // 'on' => 'create', // Limit validation to 'create' or 'update' operations
            ), 'minLength' => array('rule' => array('minLength', 20), 'required' => true, 'message' => 'Nome muito
curto'), 'maxLength' => array('rule' => array('maxLength', 150), 'required' => true, 'message' => 'Nome muito
longo'), 'Preco' => array('numeric' => array('rule' => array('decimal', 2), // 'message' => 'Your custom message
here',
                // 'allowEmpty' => false,
                // 'required' => false,
                // 'last' => false, // Stop validation after this rule
                // 'on' => 'create', // Limit validation to 'create' or 'update' operations
            ), 'usuario_id' => array('numeric' => array('rule' => array('numeric'), // 'message' => 'Your custom message here',
                // 'allowEmpty' => false,
                // 'required' => false,
                // 'last' => false, // Stop validation after this rule
                // 'on' => 'create', // Limit validation to 'create' or 'update' operations
            ), 'category_id' => array('numeric' => array('rule' => array('numeric'), // 'message' => 'Your custom message here',
                // 'allowEmpty' => false,
                // 'required' => false,
                // 'last' => false, // Stop validation after this rule
                // 'on' => 'create', // Limit validation to 'create' or 'update' operations
            ),),) [line 41]

```

Regras de Valida  es

As associa  es abaixo foram criadas com todas as chaves poss  veis, as quais n  o forem necess  rias poder  o ser removidas

- **Access public**

`void function Servico::beforeSave([$options = array()]) [line 199]`

Function Parameters:

- **\$options**

Método beforeSave

- **Access public**

void function Servico::checa_dir(\$dir, \$imagem, \$data) [line 238]

Function Parameters:

- *Array* **\$imagem**
- *String* **\$data**
- **\$dir**

Método checa_dir Verifica se o diretório existe, se não ele cria.

- **Access public**

nome function Servico::checa_nome(\$imagem, \$dir, \$data) [line 255]

Function Parameters:

- *Array* **\$imagem**
- *String* **\$data**
- **\$dir**

Método checa_nome

Verifica se o nome do arquivo já existe, se existir adiciona um numero ao nome e verifica novamente

- **Access public**

void function Servico::move_arquivos(\$imagem, \$dir, \$data) [line 291]

Function Parameters:

- *Array* **\$imagem**
- *String* **\$data**
- **\$dir**

MÃ©todo move_arquivos Move o arquivo para a pasta de destino.

- **Access public**

void function Servico::trata_nome(\$imagem_nome, \$imagem, \$data) [line 278]

Function Parameters:

- *Array* **\$imagem**
- *String* **\$data**
- **\$imagem_nome**

MÃ©todo trata_nome Trata o nome removendo espaÃ§os, acentos e caracteres em maiÃºsculo.

- **Access public**

void function Servico::upload([\$imagem = array()], [\$dir = 'img']) [line 213]

Function Parameters:

- **\$imagem**
- **\$dir**

MÃ©todo upload

Esta função permite ao usuário fazer o upload de uma imagem para o produto a ser cadastrado

- **Access** public

Appendices

Appendix A - Class Trees

Package Controller

Usuario

- [AppModel](#) (Different package)
 - [Usuario](#)

Package app-Model

AppModel

- [AppModel](#)
 - [Category](#)
 - [Comentario](#)
 - [Mensagem](#)
 - [Negociacao](#)
 - [Produto](#)
 - [Report](#)
 - [Servico](#)
 - [Usuario](#)

Package Model

Category

- [AppModel](#) (Different package)
 - [Category](#)

Comentario

- [AppModel](#) (Different package)
 - [Comentario](#)

Mensagem

- [AppModel](#) (Different package)
 - [Mensagem](#)

Negociacao

- [AppModel](#) (Different package)
 - [Negociacao](#)

Produto

- [AppModel](#) (Different package)
 - [Produto](#)

Report

- [AppModel](#) (Different package)
 - [Report](#)

Servico

- [AppModel](#) (Different package)
 - [Servico](#)

Index

A

AppModel.php	3
AppModel	1
<i>Application model for Cake.</i>	

C

Comentario	14
<i>Comentario Model</i>	
Comentario::\$belongsTo	15
<i>Associações belongsTo</i>	
Comentario::\$validate	15
<i>Regras de Validação</i>	
Comentario.php	14
Category::\$validate	13
<i>Regras de validação</i>	
Category	12
<i>Category Model</i>	
Category::\$hasMany	12
<i>hasMany associations</i>	
Category.php	11

M

Mensagem::\$validate	18
<i>Regras de validação</i>	
Mensagem	17
<i>Mensagem Model</i>	
Mensagem.php	17

N

Negociacao::\$belongsTo	22
<i>Associações belongsTo</i>	
Negociacao	21
<i>Negociacao Model</i>	
<i>As associações abaixo foram criadas com todas as chaves possíveis, as quais não forem necessárias podem ser removidas</i>	
Negociacao.php	19

P

Produto::checa_nome()	25
<i>MÃ©todo <code>checa_nome</code></i>	
Produto::move_arquivos()	25
<i>MÃ©todo <code>move_arquivos</code></i>	
<i>Move o arquivo para a pasta de destino.</i>	
Produto::trata_nome()	26
<i>MÃ©todo <code>trata_nome</code></i>	
<i>Trata o nome removendo espaÃ§os, acentos e caracteres em maiÃºsculo.</i>	
Produto::upload()	26
<i>MÃ©todo <code>upload</code></i>	
Produto::checa_dir()	25
<i>MÃ©todo <code>checa_dir</code></i>	
<i>Verifica se o diretÃ³rio existe, se nÃ£o ele cria.</i>	
Produto::beforeSave()	24
<i>MÃ©todo <code>beforeSave</code></i>	
Produto	22
<i>Produto Model</i>	
Produto::\$belongsTo	23
<i>AssociaÃ§Ãµes <code>belongsTo</code></i>	
Produto::\$hasMany	23
<i>AssociaÃ§Ãµes <code>hasMany</code></i>	
Produto::\$validate	24
<i>Regras de validaÃ§Ã£o</i>	
<i>As associaÃ§Ãµes abaixo foram criadas com todas as chaves possÃveis, as quais nÃ£o forem necessÃrias poderam ser removidas</i>	
Produto.php	20

R

Report::\$validate	28
<i>Regras de ValidaÃ§Ã£o</i>	
Report::\$belongsTo	27
<i>Associations <code>belongsTo</code></i>	
Report	27
<i>Report Model</i>	
Report.php	21

S

Servico::checa_nome()	32
<i>MÃ©todo <code>checa_nome</code></i>	
Servico::move_arquivos()	33
<i>MÃ©todo <code>move_arquivos</code></i>	
<i>Move o arquivo para a pasta de destino.</i>	
Servico::trata_nome()	33
<i>MÃ©todo <code>trata_nome</code></i>	
<i>Trata o nome removendo espaÃ§os, acentos e caracteres em maiÃºsculo.</i>	
Servico::upload()	33
<i>MÃ©todo <code>upload</code></i>	

Servico::checa_dir()	32
<i>MÃ©todo <code>checa_dir</code></i>	
<i>Verifica se o diretÃ³rio existe, se nÃ£o ele cria.</i>	
Servico::beforeSave()	31
<i>MÃ©todo <code>beforeSave</code></i>	
Servico	29
<i>Servico Model</i>	
Servico::\$belongsTo	30
<i><code>belongsTo</code> associations</i>	
Servico::\$hasMany	30
<i>AssociaÃ§Ãµes <code>hasMany</code></i>	
Servico::\$validate	31
<i>Regras de validaÃ§Ã£o</i>	
Servico.php	29

U

Usuario::move_arquivos()	7
<i>MÃ©todo <code>move_arquivos</code></i>	
<i>Move o arquivo para a pasta de destino.</i>	
Usuario::passwordConfirmation()	8
Usuario::trata_nome()	8
<i>MÃ©todo <code>trata_nome</code></i>	
<i>Trata o nome removendo espaÃ§os, acentos e caracteres em maiÃºsculo.</i>	
Usuario::upload()	8
<i><code>upload</code></i>	
<i>MÃ©todo para fazer upload de uma imagem no banco de dados.</i>	
Usuario::checa_nome()	7
<i>MÃ©todo <code>checa_nome</code></i>	
Usuario::checa_dir()	6
<i>MÃ©todo <code>checa_dir</code></i>	
<i>Verifica se o diretÃ³rio existe, se nÃ£o ele cria.</i>	
Usuario::\$hasMany	5
<i><code>hasMany</code> associations</i>	
Usuario::\$name	5
Usuario::\$validate	6
<i>Validation rules</i>	
Usuario::beforeSave()	6
<i><code>beforeSave</code></i>	
<i>Esse mÃ©todo Ã© executado antes de salvar as informaÃ§Ãµes de usuÃ¡rio.</i>	
Usuario	4