

---

# HatosGanaderos Documentation

*Publicación 1.0*

**@mrcharleon**

14 de October de 2014



<b>1. Qué es HatosGanaderos?</b>	<b>3</b>
<b>2. ¿Cómo empiezo?</b>	<b>5</b>
2.1. Instalación . . . . .	5
2.2. Tutorial 1: Crear Cuentas . . . . .	5
2.3. Tutorial 2: Configuración . . . . .	5
2.4. Tutorial 3: Técnicos . . . . .	6
2.5. Tutorial 4: Módulo de Reproducción . . . . .	6
2.6. Tutorial 5: Módulo de Producción . . . . .	6
2.7. Tutorial 6: Módulo de Sanidad . . . . .	7
2.8. Tutorial 7: Módulo de Alimentación . . . . .	7
2.9. Tutorial 8: Mensajería . . . . .	7
2.10. Tutorial 9: Reportes . . . . .	8
2.11. Tutorial 10: Notificaciones . . . . .	8
<b>3. Módulos</b>	<b>9</b>
3.1. Módulo SIDGV . . . . .	9
3.2. Módulo de Alimentos . . . . .	10
3.3. Módulo de Ganados . . . . .	12
3.4. Módulo de Medicamentos . . . . .	22
3.5. Módulo de Mensajería . . . . .	25
3.6. Módulo de Notificaciones . . . . .	26
3.7. Módulo del Perfil . . . . .	27
3.8. Módulo de Reportes . . . . .	30
3.9. Módulo Asíncrono . . . . .	31
<b>4. Índice y Tablas</b>	<b>35</b>
<b>Índice de Módulos Python</b>	<b>37</b>



HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.



---

### Qué es HatosGanaderos?

---

HatosGanaderos es un sistema web de código abierto realizado en tecnologías como: python, django, html, css, NodeJs, SPADE, etc. El software se encuentra en su primera version (1.0.0) y ya te ayuda a mantener los registros de tus ganados, esperma, alimentos, desparasitadores, vacunas, registros de producción de leche, registro de técnicos, servicio de mensajería entre técnicos y registro de técnicos.

HatosGanaderos es una excelente opción para tu ganadería, deberías probarla!





---

### ¿Cómo empiezo?

---

## 2.1 Instalación

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### 2.1.1 Windows

### 2.1.2 Linux

### 2.1.3 Mac

## 2.2 Tutorial 1: Crear Cuentas

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### 2.2.1 Registro

### 2.2.2 Recuperar Contraseña

### 2.2.3 Login

### 2.2.4 Modificar Detalles de Cuenta

### 2.2.5 Modificar Clave

### 2.2.6 Cambiar Correo Electrónico

## 2.3 Tutorial 2: Configuración

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### **2.3.1 Crear Configuración de la Ganadería**

## **2.4 Tutorial 3: Técnicos**

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### **2.4.1 Crear Cuentas de Técnicos**

### **2.4.2 Listar Cuentas de Técnicos**

## **2.5 Tutorial 4: Módulo de Reproducción**

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### **2.5.1 Registrar Ganados**

### **2.5.2 Listar ganados**

### **2.5.3 Actualizar Datos del Ganado**

### **2.5.4 Agregar Celo**

### **2.5.5 Agregar Inseminación o Monta**

### **2.5.6 Verificación de Celo**

### **2.5.7 Agregar Problemas de Gestación**

### **2.5.8 Agregar Gestación Correcta**

### **2.5.9 Registrar Esperma**

### **2.5.10 Listar Esperma**

### **2.5.11 Actualizar Datos de Esperma**

## **2.6 Tutorial 5: Módulo de Producción**

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### **2.6.1 Listar Ganados de Producción**

### **2.6.2 Agregar Ordeño**

### **2.6.3 Editar Ordeño Diario**

## **2.7 Tutorial 6: Módulo de Sanidad**

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### **2.7.1 Agregar Desparasitador**

### **2.7.2 Listar Desparasitador**

### **2.7.3 Actualizar Datos del Desparasitador**

### **2.7.4 Asignar Desparasitador a Ganados**

### **2.7.5 Agregar Vacuna**

### **2.7.6 Listar Vacuna**

### **2.7.7 Actualizar Datos de la Vacuna**

### **2.7.8 Asignar Vacuna a Ganados**

## **2.8 Tutorial 7: Módulo de Alimentación**

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### **2.8.1 Agregar Alimento**

### **2.8.2 Actualizar Alimento**

### **2.8.3 Asignar el Alimento al Ganado**

## **2.9 Tutorial 8: Mensajería**

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### **2.9.1 Enviar Mensajes**

### **2.9.2 Leer Mensajes**

## **2.10 Tutorial 9: Reportes**

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### **2.10.1 Generar un Reporte**

## **2.11 Tutorial 10: Notificaciones**

HatosGanaderos es un sistema web que te ayuda con el control y organización de reproducción, alimentación, sanidad y producción de tu entidad ganadera.

### **2.11.1 Gestionar Notificaciones**

## 3.1 Módulo SIDGV

### 3.1.1 Submodules

### 3.1.2 SIDGV.settings module

`SIDGV.settings.uggettext` (*s*)

### 3.1.3 SIDGV.settings\_dotcloud module

### 3.1.4 SIDGV.urls module

### 3.1.5 SIDGV.wsgi module

WSGI config for SIDGV\_2 project.

This module contains the WSGI application used by Django's development server and any production WSGI deployments. It should expose a module-level variable named `application`. Django's `runserver` and `runfcgi` commands discover this application via the `WSGI_APPLICATION` setting.

Usually you will have the standard Django WSGI application here, but it also might make sense to replace the whole Django WSGI application with a custom one that later delegates to the Django one. For example, you could introduce WSGI middleware here, or combine a Django application with an application of another framework.

### 3.1.6 Module contents

`SIDGV.reverse_decorator` (*func*)

## 3.2 Módulo de Alimentos

### 3.2.1 Submodules

### 3.2.2 alimentos.admin module

### 3.2.3 alimentos.forms module

```
class alimentos.forms.alimentoForm(data=None, files=None, auto_id=u'id_%s', pre-
                                fix=None, initial=None, error_class=<class 'djan-
                                go.forms.util.ErrorList'>, label_suffix=None, em-
                                pty_permitted=False, instance=None)
```

Clases base: django.forms.models.ModelForm

```
class Meta
```

```
    exclude = ['farm', 'cattle', 'status', 'is_active']
```

```
    model
```

```
        alias de Food
```

```
    widgets = {'name': <django.forms.widgets.TextInput object at 0x2b7851bcc3d0>, 'expiration_date': <django.form
```

```
alimentoForm.base_fields = {'name': <django.forms.fields.CharField object at 0x2b7851bcc510>, 'expiration_dat
```

```
alimentoForm.declared_fields = {}
```

```
alimentoForm.media
```

### 3.2.4 alimentos.models module

```
class alimentos.models.ApplicationFood(*args, **kwargs)
```

Clases base: django.db.models.base.Model

```
ApplicationFood(id, date, food_id, status)
```

```
exception DoesNotExist
```

Clases base: django.core.exceptions.ObjectDoesNotExist

```
exception ApplicationFood.MultipleObjectsReturned
```

Clases base: django.core.exceptions.MultipleObjectsReturned

```
ApplicationFood.STATUS_CHOICES = ((0, 'Realizado'), (1, 'Cancelado'))
```

```
ApplicationFood.cattle
```

```
ApplicationFood.food
```

```
ApplicationFood.get_next_by_date(*moreargs, **morekwargs)
```

```
ApplicationFood.get_previous_by_date(*moreargs, **morekwargs)
```

```
ApplicationFood.get_status_display(*moreargs, **morekwargs)
```

```
ApplicationFood.objects = <django.db.models.manager.Manager object at 0x2b785178e0d0>
```

```
class alimentos.models.Food(*args, **kwargs)
```

Clases base: django.db.models.base.Model

Food(id, name, expiration\_date, unit, amount, sex, farm\_id, phase, consumer\_amount, interval, time\_interval, administration\_route, observations, is\_active)

**ADMINISTRATION\_ROUTE\_CHOICES** = ((0, 'Oral'), (1, 'Granulada'))

**exception DoesNotExist**

Clases base: django.core.exceptions.ObjectDoesNotExist

**exception Food.MultipleObjectsReturned**

Clases base: django.core.exceptions.MultipleObjectsReturned

Food.**PHASE\_CHOICES** = ((0, 'Ternera(o)'), (1, 'Vacona'), (2, 'Vientre'), (3, 'Ternera(o) y Vacona'), (4, 'Ternera(o) y Vie'))

Food.**SEX\_CHOICES** = ((0, 'Hembra'), (1, 'Macho'), (2, 'Hembra y Macho'))

Food.**TIME\_INTERVAL\_CHOICES** = ((0, u'Intervalo en d\xedas'), (1, 'Intervalo en meses'), (2, u'Intervalo en a\xf1os'))

Food.**UNIT\_CHOICES** = ((0, 'ml'), (1, 'gr'), (2, 'lbs'), (3, 'Kg'), (4, 'Paquetes'))

Food.**application\_food\_cattle**

Food.**farm**

Food.**get\_administration\_route\_display** (\*moreargs, \*\*morekwargs)

Food.**get\_next\_by\_expiration\_date** (\*moreargs, \*\*morekwargs)

Food.**get\_phase\_display** (\*moreargs, \*\*morekwargs)

Food.**get\_previous\_by\_expiration\_date** (\*moreargs, \*\*morekwargs)

Food.**get\_sex\_display** (\*moreargs, \*\*morekwargs)

Food.**get\_time\_interval\_display** (\*moreargs, \*\*morekwargs)

Food.**get\_unit\_display** (\*moreargs, \*\*morekwargs)

Food.**notification\_food**

Food.**objects** = <django.db.models.manager.Manager object at 0x2b7851781490>

### 3.2.5 alimentos.tests module

### 3.2.6 alimentos.views module

alimentos.views.**add\_food** (request, \*args, \*\*kwargs)

alimentos.views.**asigna\_alimento** (request, \*args, \*\*kwargs)

alimentos.views.**edit\_food** (request, \*args, \*\*kwargs)

alimentos.views.**list\_food** (request, \*args, \*\*kwargs)

### 3.2.7 Module contents

## 3.3 Módulo de Ganados

### 3.3.1 Submodules

#### 3.3.2 ganados.admin module

#### 3.3.3 ganados.forms module

```
class ganados.forms.attemptForm(data=None, files=None, auto_id=u'id_%s', prefix=None,
                                initial=None, error_class=<class 'django.forms.util.ErrorList'>,
                                label_suffix=None, empty_permitted=False, instance=None)
```

Clases base: django.forms.models.ModelForm

**class Meta**

**exclude** = ['attempt', 'attempt\_date', 'state', 'verification']

**model**

alias de Attempt

**widgets** = {'observations': <django.forms.widgets.Textarea object at 0x2b7851dba7d0>, 'rp\_father': <django.form

attemptForm.**base\_fields** = {'observations': <django.forms.fields.CharField object at 0x2b7851dba8d0>, 'type\_con

attemptForm.**declared\_fields** = {}

attemptForm.**media**

```
class ganados.forms.attemptServiceForm(data=None, files=None, auto_id=u'id_%s', pre-
                                         fix=None, initial=None, error_class=<class 'djan-
                                         go.forms.util.ErrorList'>, label_suffix=None, em-
                                         pty_permitted=False, instance=None)
```

Clases base: django.forms.models.ModelForm

**class Meta**

**exclude** = ['verification']

**model**

alias de Attempt

attemptServiceForm.**base\_fields** = {'attempt': <django.forms.fields.IntegerField object at 0x2b7851dbae50>, 'at

attemptServiceForm.**declared\_fields** = {}

attemptServiceForm.**media**

```
class ganados.forms.editaGanadoCeloForm(data=None, files=None, auto_id=u'id_%s', pre-
                                          fix=None, initial=None, error_class=<class 'djan-
                                          go.forms.util.ErrorList'>, label_suffix=None, em-
                                          pty_permitted=False, instance=None)
```

Clases base: django.forms.models.ModelForm

**class Meta**

**exclude** = ['ganado', 'is\_active', 'fecha\_fin', 'estado']



```

    model
        alias de Celo

    widgets = {'observaciones': <django.forms.widgets.Textarea object at 0x2b7851dba610>, 'fecha_inicio': <django.f

editaGanadoCeloForm.base_fields = {'fecha_inicio': <django.forms.fields.DateTimeField object at 0x2b7851dba
editaGanadoCeloForm.declared_fields = {}
editaGanadoCeloForm.media

class ganados.forms.etapaForm(data=None, files=None, auto_id=u'id_%s', prefix=None,
                               initial=None, error_class=<class 'django.forms.util.ErrorList'>,
                               label_suffix=None, empty_permitted=False, instance=None)
Clases base: django.forms.models.ModelForm

class Meta

    exclude = ['is_active']

    model
        alias de Etapa

    etapaForm.base_fields = {'fecha_inicio': <django.forms.fields.DateField object at 0x2b7851db3c50>, 'nombre': <d
    etapaForm.declared_fields = {}
    etapaForm.media

class ganados.forms.ganadoForm(data=None, files=None, auto_id=u'id_%s', prefix=None,
                                initial=None, error_class=<class 'django.forms.util.ErrorList'>,
                                label_suffix=None, empty_permitted=False, instance=None)
Clases base: django.forms.models.ModelForm

class Meta

    exclude = ['ganaderia', 'identificacion_simple', 'identificacion_ecuador', 'edad', 'etapa', 'verificacion', 'celo', 'cicl

    model
        alias de Ganado

    widgets = {'observaciones': <django.forms.widgets.Textarea object at 0x2b7851db3fd0>, 'nacimiento': <django.f

ganadoForm.base_fields = {'imagen': <django.forms.fields.ImageField object at 0x2b7851dba150>, 'nacimiento': <d
ganadoForm.declared_fields = {}
ganadoForm.media

class ganados.forms.gestacionForm(data=None, files=None, auto_id=u'id_%s', pre-
                                  fix=None, initial=None, error_class=<class 'djan-
                                  go.forms.util.ErrorList'>, label_suffix=None, em-
                                  pty_permitted=False, instance=None)
Clases base: django.forms.models.ModelForm

class Meta

    exclude = ['problema', 'is_active', 'ganado']

    model
        alias de Gestacion

    widgets = {'fecha_servicio': <django.forms.widgets.TextInput object at 0x2b7851dc7290>, 'observaciones': <djang

```

```

gestacionForm.base_fields = {'fecha_servicio': <django.forms.fields.DateField object at 0x2b7851dc7610>, 'fecha
gestacionForm.declared_fields = {}
gestacionForm.media

class ganados.forms.inseminationForm(data=None, files=None, auto_id=u'id_%s', pre-
fix=None, initial=None, error_class=<class 'djan-
go.forms.util.ErrorList'>, label_suffix=None, em-
pty_permitted=False, instance=None)
Clases base: django.forms.models.ModelForm

class Meta

    exclude = ['down_insemination', 'farm', 'rp']

    model
        alias de Insemination

    widgets = {'observations': <django.forms.widgets.Textarea object at 0x2b7851d43bd0>, 'registration_date': <djan

inseminationForm.base_fields = {'name': <django.forms.fields.CharField object at 0x2b7851d43d10>, 'registrati
inseminationForm.declared_fields = {}
inseminationForm.media

class ganados.forms.ordenioForm(data=None, files=None, auto_id=u'id_%s', prefix=None,
initial=None, error_class=<class 'django.forms.util.ErrorList'>,
label_suffix=None, empty_permitted=False, instance=None)
Clases base: django.forms.models.ModelForm

class Meta

    exclude = ['total', 'numero_ordenio', 'ganado', 'fecha']

    model
        alias de Ordenio

    widgets = {'observaciones': <django.forms.widgets.Textarea object at 0x2b7851db39d0>, 'cantidad': <django.form

ordenioForm.base_fields = {'cantidad': <django.forms.fields.IntegerField object at 0x2b7851db3ad0>, 'observacio
ordenioForm.declared_fields = {}
ordenioForm.media

class ganados.forms.problemGestacionForm(data=None, files=None, auto_id=u'id_%s', pre-
fix=None, initial=None, error_class=<class 'djan-
go.forms.util.ErrorList'>, label_suffix=None, em-
pty_permitted=False, instance=None)
Clases base: django.forms.models.ModelForm

class Meta

    model
        alias de ProblemaGestacion

    widgets = {'fecha_problema': <django.forms.widgets.TextInput object at 0x2b7851dc75d0>, 'observaciones': <dja

problemGestacionForm.base_fields = {'fecha_problema': <django.forms.fields.DateField object at 0x2b7851dc7
problemGestacionForm.declared_fields = {}

```

```

problemGestacionForm.media

class ganados.forms.tipoNormaEcuadorForm(data=None, files=None, auto_id=u'id_%s', pre-
                                         fix=None, initial=None, error_class=<class 'djan-
                                         go.forms.util.ErrorList'>, label_suffix=None, em-
                                         pty_permitted=False, instance=None)
Clases base: django.forms.models.ModelForm

class Meta

    exclude = ['rp']

    model
        alias de Identificacion_Ecuador

    widgets = {'rp_padre': <django.forms.widgets.TextInput object at 0x2b7851db33d0>, 'codigo_pais': <django.form

tipoNormaEcuadorForm.base_fields = {'siglas_pais': <django.forms.fields.CharField object at 0x2b7851db3510>}
tipoNormaEcuadorForm.declared_fields = {}
tipoNormaEcuadorForm.media

class ganados.forms.tipoSimpleForm(data=None, files=None, auto_id=u'id_%s', pre-
                                   fix=None, initial=None, error_class=<class 'djan-
                                   go.forms.util.ErrorList'>, label_suffix=None, em-
                                   pty_permitted=False, instance=None)
Clases base: django.forms.models.ModelForm

class Meta

    exclude = ['rp']

    model
        alias de Identificacion_Simple

    widgets = {'nombre': <django.forms.widgets.TextInput object at 0x2b7851d43c50>, 'rp_padre': <django.forms.wi

tipoSimpleForm.base_fields = {'nombre': <django.forms.fields.CharField object at 0x2b7851db3050>, 'rp_madr

tipoSimpleForm.declared_fields = {}
tipoSimpleForm.media

class ganados.forms.verificationForm(data=None, files=None, auto_id=u'id_%s', pre-
                                     fix=None, initial=None, error_class=<class 'djan-
                                     go.forms.util.ErrorList'>, label_suffix=None, em-
                                     pty_permitted=False, instance=None)
Clases base: django.forms.models.ModelForm

class Meta

    model
        alias de Verification

verificationForm.base_fields = {'initial_date': <django.forms.fields.DateField object at 0x2b7851dbaad0>, 'is_a

verificationForm.declared_fields = {}
verificationForm.media

```

```
class ganados.forms.verifyAttemptForm(data=None, files=None, auto_id=u'id_%s', pre-
                                     fix=None, initial=None, error_class=<class 'djan-
                                     go.forms.util.ErrorList'>, label_suffix=None, em-
                                     pty_permitted=False, instance=None)
```

Clases base: django.forms.models.ModelForm

**class Meta**

**exclude** = ['attempt', 'attempt\_date', 'verification']

**model**

alias de Attempt

**widgets** = {'observations': <django.forms.widgets.Textarea object at 0x2b7851dc7210>, 'rp\_father': <django.form

verifyAttemptForm.**base\_fields** = {'state': <django.forms.fields.TypedChoiceField object at 0x2b7851dc7310>, 'c

verifyAttemptForm.**declared\_fields** = {}

verifyAttemptForm.**media**

### 3.3.4 ganados.models module

```
class ganados.models.Attempt(*args, **kwargs)
```

Clases base: django.db.models.base.Model

Attempt(id, attempt, attempt\_date, state, observations, type\_conception, rp\_father, verification\_id)

**exception DoesNotExist**

Clases base: django.core.exceptions.ObjectDoesNotExist

**exception Attempt.MultipleObjectsReturned**

Clases base: django.core.exceptions.MultipleObjectsReturned

Attempt.**STATE\_CHOICES** = ((0, 'Correcto'), (1, 'Incorrecto'))

Attempt.**TYPE\_CONCEPTION\_CHOICES** = ((0, u'Inseminaci\xf3n'), (1, 'Monta'))

Attempt.**get\_next\_by\_attempt\_date**(\*moreargs, \*\*morekwargs)

Attempt.**get\_previous\_by\_attempt\_date**(\*moreargs, \*\*morekwargs)

Attempt.**get\_state\_display**(\*moreargs, \*\*morekwargs)

Attempt.**get\_type\_conception\_display**(\*moreargs, \*\*morekwargs)

Attempt.**objects** = <django.db.models.manager.Manager object at 0x2b785175b390>

Attempt.**verification**

```
class ganados.models.Celo(*args, **kwargs)
```

Clases base: django.db.models.base.Model

Celo(id, fecha\_inicio, fecha\_fin, estado, observaciones, ganado\_id, is\_active)

**exception DoesNotExist**

Clases base: django.core.exceptions.ObjectDoesNotExist

Celo.**ESTADO\_CHOICES** = ((0, 'En celo'), (1, 'Sin celo'))

**exception Celo.MultipleObjectsReturned**

Clases base: django.core.exceptions.MultipleObjectsReturned

Celo.**ganado**

```

Celo.get_estado_display(*moreargs, **morekwargs)
Celo.get_next_by_fecha_fin(*moreargs, **morekwargs)
Celo.get_next_by_fecha_inicio(*moreargs, **morekwargs)
Celo.get_previous_by_fecha_fin(*moreargs, **morekwargs)
Celo.get_previous_by_fecha_inicio(*moreargs, **morekwargs)
Celo.objects = <django.db.models.manager.Manager object at 0x2b785175be90>
class ganados.models.Ciclo(*args, **kwargs)
    Clases base: django.db.models.base.Model
    Ciclo(id, fecha_inicio, nombre, fecha_fin, ganado_id, is_active)
    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist
    exception Ciclo.MultipleObjectsReturned
        Clases base: django.core.exceptions.MultipleObjectsReturned
    Ciclo.NOMBRE_CHOICES = ((0, u'Per\xedodo Vac\xedo'), (1, u'Per\xedodo seco'), (2, u'Per\xedodo lactancia'), (3, u'Per\xedodo
    Ciclo.ganado
    Ciclo.get_next_by_fecha_fin(*moreargs, **morekwargs)
    Ciclo.get_next_by_fecha_inicio(*moreargs, **morekwargs)
    Ciclo.get_nombre_display(*moreargs, **morekwargs)
    Ciclo.get_previous_by_fecha_fin(*moreargs, **morekwargs)
    Ciclo.get_previous_by_fecha_inicio(*moreargs, **morekwargs)
    Ciclo.objects = <django.db.models.manager.Manager object at 0x2b7851765350>
class ganados.models.DownCattle(*args, **kwargs)
    Clases base: django.db.models.base.Model
    DownCattle(id, date, cause_down, observations)
    CAUSE_DOWN_CHOICES = ((0, 'Muerte'), (1, 'Venta'), (2, 'Desaparici\x3\x3n'))
    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist
    exception DownCattle.MultipleObjectsReturned
        Clases base: django.core.exceptions.MultipleObjectsReturned
    DownCattle.cattle_down_cattle
    DownCattle.get_cause_down_display(*moreargs, **morekwargs)
    DownCattle.get_next_by_date(*moreargs, **morekwargs)
    DownCattle.get_previous_by_date(*moreargs, **morekwargs)
    DownCattle.objects = <django.db.models.manager.Manager object at 0x2b7851740ad0>
class ganados.models.DownInsemination(*args, **kwargs)
    Clases base: django.db.models.base.Model
    DownInsemination(id, date, cause_down, observations)
    CAUSE_DOWN_CHOICES = ((0, 'Agotamiento'), (1, u'Muestra inv\xellida'))

```

```

exception DoesNotExist
    Clases base: django.core.exceptions.ObjectDoesNotExist

exception DownInsemination.MultipleObjectsReturned
    Clases base: django.core.exceptions.MultipleObjectsReturned

DownInsemination.get_cause_down_display (*moreargs, **morekwargs)

DownInsemination.get_next_by_date (*moreargs, **morekwargs)

DownInsemination.get_previous_by_date (*moreargs, **morekwargs)

DownInsemination.insemination_down

DownInsemination.objects = <django.db.models.manager.Manager object at 0x2b7851771610>

class ganados.models.Etapa (*args, **kwargs)
    Clases base: django.db.models.base.Model

    Etapa(id, fecha_inicio, nombre, observaciones, ganado_id, is_active)

exception DoesNotExist
    Clases base: django.core.exceptions.ObjectDoesNotExist

exception Etapa.MultipleObjectsReturned
    Clases base: django.core.exceptions.MultipleObjectsReturned

Etapa.NOMBRE_CHOICES = ((0, 'Ternera'), (1, 'Vacona'), (2, 'Vientre'))

Etapa.ganado

Etapa.get_next_by_fecha_inicio (*moreargs, **morekwargs)

Etapa.get_nombre_display (*moreargs, **morekwargs)

Etapa.get_previous_by_fecha_inicio (*moreargs, **morekwargs)

Etapa.objects = <django.db.models.manager.Manager object at 0x2b7851771310>

class ganados.models.Ganado (*args, **kwargs)
    Clases base: django.db.models.base.Model

    Ganado(id, imagen, ganaderia_id, nacimiento, genero, raza, forma_concepcion, live_weight, unit_live_weight,
    observaciones, edad_anios, edad_meses, edad_dias, identificacion_simple_id, identificacion_ecuador_id,
    down_cattle_id)

    CONCEPCION_CHOICES = ((0, u'Inseminaci\xf3n'), (1, 'Monta'))

exception DoesNotExist
    Clases base: django.core.exceptions.ObjectDoesNotExist

Ganado.GENDER_CHOICES = ((0, 'Macho'), (1, 'Hembra'), (2, 'Sin Definir'))

exception Ganado.MultipleObjectsReturned
    Clases base: django.core.exceptions.MultipleObjectsReturned

Ganado.RAZAS_CHOICES = ((0, 'Angus'), (1, 'Ankole'), (2, 'Asturiana de los Valles'), (3, 'Avenia'), (4, 'Blonde D Aquilón'))

Ganado.UNIT_LIVE_WEIGHT_CHOICES = ((0, 'Kg'), (1, 'lbs'), (2, 'Arrobas'))

Ganado.application_food_food

Ganado.application_medicament_medicament

Ganado.celos

Ganado.ciclos

```

```

Ganado.down_cattle
Ganado.etapas
Ganado.ganaderia
Ganado.gestaciones
Ganado.get_forma_concepcion_display(*moreargs, **morekwargs)
Ganado.get_genero_display(*moreargs, **morekwargs)
Ganado.get_next_by_nacimiento(*moreargs, **morekwargs)
Ganado.get_previous_by_nacimiento(*moreargs, **morekwargs)
Ganado.get_raza_display(*moreargs, **morekwargs)
Ganado.get_unit_live_weight_display(*moreargs, **morekwargs)
Ganado.identificacion_ecuador
Ganado.identificacion_simple
Ganado.notification_cattle
Ganado.objects = <django.db.models.manager.Manager object at 0x2b785174c8d0>
Ganado.ordenios
Ganado.verification_cattle

class ganados.models.Gestacion(*args, **kwargs)
    Clases base: django.db.models.base.Model

    Gestacion(id, fecha_servicio, fecha_parto, tipo_parto, observaciones, problema_id, ganado_id, is_active)

    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist

    exception Gestacion.MultipleObjectsReturned
        Clases base: django.core.exceptions.MultipleObjectsReturned

    Gestacion.TIPO_PARTO_CHOICES = ((0, 'Natural'), (1, u'Ces\xe1rea'))

    Gestacion.ganado

    Gestacion.get_next_by_fecha_parto(*moreargs, **morekwargs)
    Gestacion.get_next_by_fecha_servicio(*moreargs, **morekwargs)
    Gestacion.get_previous_by_fecha_parto(*moreargs, **morekwargs)
    Gestacion.get_previous_by_fecha_servicio(*moreargs, **morekwargs)
    Gestacion.get_tipo_parto_display(*moreargs, **morekwargs)
    Gestacion.objects = <django.db.models.manager.Manager object at 0x2b7851765e50>
    Gestacion.problema

class ganados.models.Identificacion_Ecuador(*args, **kwargs)
    Clases base: django.db.models.base.Model

    Identificacion_Ecuador(id, siglas_pais, codigo_pais, codigo_provincia, numero_serie, codigo_barras, rp, nombre, rp_madre, rp_padre)

    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist

```

```

exception Identificacion_Ecuador.MultipleObjectsReturned
    Clases base: django.core.exceptions.MultipleObjectsReturned

Identificacion_Ecuador.identificaciones_ecuador

Identificacion_Ecuador.objects = <django.db.models.manager.Manager object at 0x2b7851740750>

class ganados.models.Identificacion_Simple (*args, **kwargs)
    Clases base: django.db.models.base.Model

    Identificacion_Simple(id, rp, nombre, rp_madre, rp_padre)

exception DoesNotExist
    Clases base: django.core.exceptions.ObjectDoesNotExist

exception Identificacion_Simple.MultipleObjectsReturned
    Clases base: django.core.exceptions.MultipleObjectsReturned

Identificacion_Simple.identificaciones_simples

Identificacion_Simple.objects = <django.db.models.manager.Manager object at 0x2b7851740110>

class ganados.models.Insemination (*args, **kwargs)
    Clases base: django.db.models.base.Model

    Insemination(id, down_insemination_id, farm_id, rp, name, registration_date, amount_pajuelas, breed, obser-
    vations)

BREED_CHOICES = ((0, 'Angus'), (1, 'Ankole'), (2, 'Asturiana de los Valles'), (3, 'Avilenia'), (4, 'Blonde D Aquitaine'), (5,

exception DoesNotExist
    Clases base: django.core.exceptions.ObjectDoesNotExist

exception Insemination.MultipleObjectsReturned
    Clases base: django.core.exceptions.MultipleObjectsReturned

Insemination.down_insemination

Insemination.farm

Insemination.get_breed_display (*moreargs, **morekwargs)

Insemination.get_next_by_registration_date (*moreargs, **morekwargs)

Insemination.get_previous_by_registration_date (*moreargs, **morekwargs)

Insemination.objects = <django.db.models.manager.Manager object at 0x2b7851771cd0>

class ganados.models.Ordenio (*args, **kwargs)
    Clases base: django.db.models.base.Model

    Ordenio(id, fecha, numero_ordenio, cantidad, total, observaciones, ganado_id)

exception DoesNotExist
    Clases base: django.core.exceptions.ObjectDoesNotExist

exception Ordenio.MultipleObjectsReturned
    Clases base: django.core.exceptions.MultipleObjectsReturned

Ordenio.ganado

Ordenio.get_next_by_fecha (*moreargs, **morekwargs)

Ordenio.get_previous_by_fecha (*moreargs, **morekwargs)

Ordenio.objects = <django.db.models.manager.Manager object at 0x2b785175b910>

```



```

class ganados.models.ProblemaGestacion(*args, **kwargs)
    Clases base: django.db.models.base.Model

    ProblemaGestacion(id, fecha_problema, tipo_problema, observaciones)

    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist

    exception ProblemaGestacion.MultipleObjectsReturned
        Clases base: django.core.exceptions.MultipleObjectsReturned

    ProblemaGestacion.TIPO_PROBLEMA_CHOICES = ((0, 'Aborto'), (1, 'Nacido muerto'), (2, 'Madre muerta'), (3, 'Lo

    ProblemaGestacion.gestacion

    ProblemaGestacion.get_next_by_fecha_problema(*moreargs, **morekwargs)

    ProblemaGestacion.get_previous_by_fecha_problema(*moreargs, **morekwargs)

    ProblemaGestacion.get_tipo_problema_display(*moreargs, **morekwargs)

    ProblemaGestacion.objects = <django.db.models.manager.Manager object at 0x2b7851765710>

class ganados.models.Verification(*args, **kwargs)
    Clases base: django.db.models.base.Model

    Verification(id, initial_date, is_active, cattle_id)

    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist

    exception Verification.MultipleObjectsReturned
        Clases base: django.core.exceptions.MultipleObjectsReturned

    Verification.attempt_verification

    Verification.cattle

    Verification.get_next_by_initial_date(*moreargs, **morekwargs)

    Verification.get_previous_by_initial_date(*moreargs, **morekwargs)

    Verification.objects = <django.db.models.manager.Manager object at 0x2b785174cd10>

```

### 3.3.5 ganados.tests module

### 3.3.6 ganados.views module

```

ganados.views.add_attempt_service(request, *args, **kwargs)

ganados.views.add_cattle(request, *args, **kwargs)

ganados.views.add_insemination(request, *args, **kwargs)

ganados.views.add_service(request, *args, **kwargs)

ganados.views.agrega_ganado_ordenio(request, username, ganado_id)

ganados.views.calcula_edad_anios(request, date)

ganados.views.calcula_edad_dias(request, date)

ganados.views.calcula_edad_meses(request, date)

ganados.views.calcula_etapa(request, anios, meses, etapa_tenera, etapa_vacona)

```

```

ganados.views.edit_cattle_male(request, *args, **kwargs)
ganados.views.edit_insemination(request, *args, **kwargs)
ganados.views.edita_ganado(request, *args, **kwargs)
ganados.views.edita_ganado_celo(request, *args, **kwargs)
ganados.views.edita_ganado_ordenio(request, username, ganado_id, num_ordenio)
ganados.views.gestacion(request, *args, **kwargs)
ganados.views.list_cattle(request, *args, **kwargs)
ganados.views.list_cattle_male(request, *args, **kwargs)
ganados.views.list_insemination(request, *args, **kwargs)
ganados.views.lista_ganado_produccion(request, *args, **kwargs)
ganados.views.problem_gestacion(request, *args, **kwargs)
ganados.views.verify_attempt(request, *args, **kwargs)

```

### 3.3.7 Module contents

## 3.4 Módulo de Medicamentos

### 3.4.1 Submodules

### 3.4.2 medicament.admin module

### 3.4.3 medicament.forms module

```

class medicament.forms.vaccineForm(data=None, files=None, auto_id=u'id_%s', pre-
    fix=None, initial=None, error_class=<class 'djan-
    go.forms.util.ErrorList'>, label_suffix=None, em-
    pty_permitted=False, instance=None)

```

Clases base: django.forms.models.ModelForm

**class Meta**

**exclude** = ['farm', 'is\_vaccine', 'status', 'is\_wormer', 'cattle']

**model**

alias de Medicament

**widgets** = {'amount': <django.forms.widgets.TextInput object at 0x2b785292ce90>, 'name': <django.forms.widget

vaccineForm.**base\_fields** = {'name': <django.forms.fields.CharField object at 0x2b785292cc50>, 'expiration\_date'

vaccineForm.**declared\_fields** = {}

vaccineForm.**media**

```

class medicament.forms.wormerForm(data=None, files=None, auto_id=u'id_%s', pre-
    fix=None, initial=None, error_class=<class 'djan-
    go.forms.util.ErrorList'>, label_suffix=None, em-
    pty_permitted=False, instance=None)

```

Clases base: django.forms.models.ModelForm

**class Meta**

```

    exclude = ['farm', 'is_vaccine', 'status', 'is_wormer', 'cattle']

    model
        alias de Medicament

    widgets = {'amount': <django.forms.widgets.TextInput object at 0x2b7852779e10>, 'name': <django.forms.widget
wormerForm.base_fields = {'name': <django.forms.fields.CharField object at 0x2b78520f4b10>, 'expiration_date':
wormerForm.declared_fields = {}
wormerForm.media

```

### 3.4.4 medicament.models module

```

class medicament.models.ApplicationMedicament (*args, **kwargs)
    Clases base: django.db.models.base.Model

    ApplicationMedicament(id, date, medicament_id, status)

    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist

    exception ApplicationMedicament.MultipleObjectsReturned
        Clases base: django.core.exceptions.MultipleObjectsReturned

    ApplicationMedicament.STATUS_CHOICES = ((0, 'Realizado'), (1, 'Cancelado'))

    ApplicationMedicament.cattle

    ApplicationMedicament.get_next_by_date (*moreargs, **morekwargs)

    ApplicationMedicament.get_previous_by_date (*moreargs, **morekwargs)

    ApplicationMedicament.get_status_display (*moreargs, **morekwargs)

    ApplicationMedicament.medicament

    ApplicationMedicament.objects = <django.db.models.manager.Manager object at 0x2b785179a9d0>

class medicament.models.Medicament (*args, **kwargs)
    Clases base: django.db.models.base.Model

    Medicament(id, name, expiration_date, unit, amount, sex, farm_id, application_age, time_application_age,
amount_application, option_number_application, number_application, interval, time_interval, administra-
tion_route, observations, is_vaccine, is_wormer, is_active)

    ADMINISTRATION_ROUTE_CHOICES = ((0, 'Intravenosa'), (1, 'Intramuscular'), (2, u'Subcut\xea1nea'), (3, 'Intraperiton

    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist

    exception Medicament.MultipleObjectsReturned
        Clases base: django.core.exceptions.MultipleObjectsReturned

    Medicament.OPTION_NUMBER_APPLICATION = ((0, 'Veces exactas'), (1, 'Repetitivo'))

    Medicament.SEX_CHOICES = ((0, 'Hembra'), (1, 'Macho'), (2, 'Hembra y Macho'))

    Medicament.TIME_APPLICATION_AGE_CHOICES = ((0, u'Aplicaci\xf3n en d\xedas'), (1, 'Aplicaci\xf3n en meses

    Medicament.TIME_INTERVAL_CHOICES = ((0, u'Intervalo en d\xedas'), (1, 'Intervalo en meses'), (2, u'Intervalo en a

```

```
Medicament.UNIT_CHOICES = ((0, 'ml'), (1, 'gr'), (2, 'lbs'), (3, 'Kg'), (4, 'Paquetes'))
Medicament.application_medicament_cattle
Medicament.farm
Medicament.get_administration_route_display(*moreargs, **morekwargs)
Medicament.get_next_by_expiration_date(*moreargs, **morekwargs)
Medicament.get_option_number_application_display(*moreargs, **morekwargs)
Medicament.get_previous_by_expiration_date(*moreargs, **morekwargs)
Medicament.get_sex_display(*moreargs, **morekwargs)
Medicament.get_time_application_age_display(*moreargs, **morekwargs)
Medicament.get_time_interval_display(*moreargs, **morekwargs)
Medicament.get_unit_display(*moreargs, **morekwargs)
Medicament.notification_medicament
Medicament.objects = <django.db.models.manager.Manager object at 0x2b785178eed0>
```

### 3.4.5 medicament.tests module

### 3.4.6 medicament.views module

```
medicament.views.add_vaccine(request, *args, **kwargs)
medicament.views.add_wormer(request, *args, **kwargs)
medicament.views.assign_vaccine(request, *args, **kwargs)
medicament.views.assign_wormer(request, *args, **kwargs)
medicament.views.edit_vaccine(request, *args, **kwargs)
medicament.views.edit_wormer(request, *args, **kwargs)
medicament.views.list_vaccine(request, *args, **kwargs)
medicament.views.list_wormer(request, *args, **kwargs)
```

### 3.4.7 Module contents

## 3.5 Módulo de Mensajería

### 3.5.1 Submodules

### 3.5.2 messages.admin module

### 3.5.3 messages.forms module

```
class messages.forms.MessageForm(data=None, files=None, auto_id=u'id_%s', pre-
                                fix=None, initial=None, error_class=<class 'djan-
                                go.forms.util.ErrorList'>, label_suffix=None, em-
                                pty_permitted=False, instance=None)
```

Clases base: django.forms.models.ModelForm

**class Meta**

**exclude** = ['sender', 'sent\_at', 'read\_at', 'front']

**model**

alias de Message

**widgets** = {'content': <django.forms.widgets.Textarea object at 0x2b784c680950>}

MessageForm.**base\_fields** = {'receiver': <django.forms.models.ModelChoiceField object at 0x2b7852946b90>, 'con

MessageForm.**declared\_fields** = {}

MessageForm.**media**

```
class messages.forms.MessageResponseForm(data=None, files=None, auto_id=u'id_%s', pre-
                                          fix=None, initial=None, error_class=<class 'djan-
                                          go.forms.util.ErrorList'>, label_suffix=None, em-
                                          pty_permitted=False, instance=None)
```

Clases base: django.forms.models.ModelForm

**class Meta**

**exclude** = ['sender', 'sent\_at', 'read\_at', 'front', 'receiver']

**model**

alias de Message

**widgets** = {'content': <django.forms.widgets.Textarea object at 0x2b7852946c50>}

MessageResponseForm.**base\_fields** = {'content': <django.forms.fields.CharField object at 0x2b7852779c50>}

MessageResponseForm.**declared\_fields** = {}

MessageResponseForm.**media**

### 3.5.4 messages.models module

```
class messages.models.Message(*args, **kwargs)
```

Clases base: django.db.models.base.Model

Message(id, sender\_id, receiver\_id, content, sent\_at, read\_at, front)

#### **exception DoesNotExist**

Clases base: `django.core.exceptions.ObjectDoesNotExist`

#### **exception Message.MultipleObjectsReturned**

Clases base: `django.core.exceptions.MultipleObjectsReturned`

`Message.get_next_by_sent_at (*moreargs, **morekwargs)`

`Message.get_previous_by_sent_at (*moreargs, **morekwargs)`

`Message.objects = <django.db.models.manager.Manager object at 0x2b78517d8510>`

`Message.receiver`

`Message.sender`

### 3.5.5 messages.tests module

### 3.5.6 messages.views module

`messages.views.alert_messages (request)`

`messages.views.messages_details (request, user_id, user_send_id, user_receiver_id)`

`messages.views.messages_list (request)`

`messages.views.messages_list_no_read (request)`

`messages.views.messages_list_read (request)`

`messages.views.new_message (request)`

### 3.5.7 Module contents

## 3.6 Módulo de Notificaciones

### 3.6.1 Submodules

### 3.6.2 notifications.admin module

### 3.6.3 notifications.models module

**class** `notifications.models.Notification (*args, **kwargs)`

Clases base: `django.db.models.base.Model`

`Notification(id, start_date, end_date, state, module, ident_cattle_id, ident_medicament_id, ident_food_id, name)`

#### **exception DoesNotExist**

Clases base: `django.core.exceptions.ObjectDoesNotExist`

`Notification.MODULE_CHOICES = ((0, u'Reproducci\xf3n'), (1, u'Alimentaci\xf3n'), (2, 'Sanidad'), (3, u'Producci\xf3n'))`

#### **exception Notification.MultipleObjectsReturned**

Clases base: `django.core.exceptions.MultipleObjectsReturned`

`Notification.NAME_CHOICES = ((0, 'Ganado en celo'), (1, 'Registro del servicio'), (2, u'Verificaci\xf3n del celo'), (3, 'Pendiente'))`

`Notification.STATE_CHOICES = ((0, 'No realizado'), (1, 'Realizado'), (2, 'Pendiente'))`

```

Notification.get_module_display(*moreargs, **morekwargs)
Notification.get_name_display(*moreargs, **morekwargs)
Notification.get_next_by_end_date(*moreargs, **morekwargs)
Notification.get_next_by_start_date(*moreargs, **morekwargs)
Notification.get_previous_by_end_date(*moreargs, **morekwargs)
Notification.get_previous_by_start_date(*moreargs, **morekwargs)
Notification.get_state_display(*moreargs, **morekwargs)
Notification.ident_cattle
Notification.ident_food
Notification.ident_medicament
Notification.objects = <django.db.models.manager.Manager object at 0x2b78517a8210>

```

### 3.6.4 notifications.tests module

### 3.6.5 notifications.views module

```

notifications.views.list_notifications(request, *args, **kwargs)
notifications.views.list_notifications_alimentacion(request, *args, **kwargs)
notifications.views.list_notifications_produccion(request, *args, **kwargs)
notifications.views.list_notifications_reproduccion(request, *args, **kwargs)
notifications.views.list_notifications_sanidad(request, *args, **kwargs)
notifications.views.realizedNotification(request, *args, **kwargs)

```

### 3.6.6 Module contents

## 3.7 Módulo del Perfil

### 3.7.1 Submodules

### 3.7.2 profiles.admin module

### 3.7.3 profiles.forms module

```

class profiles.forms.ConfiguracionForm(data=None, files=None, auto_id=u'id_%s', pre-
                                     fix=None, initial=None, error_class=<class 'djan-
                                     go.forms.util.ErrorList'>, label_suffix=None, em-
                                     pty_permitted=False, instance=None)

```

Clases base: django.forms.models.ModelForm

class Meta

```
exclude = ['etapa_vientre']
```

```

    model
        alias de Configuracion

    widgets = {'etapa_tenera': <django.forms.widgets.TextInput object at 0x2b7851d43a50>, 'celo_duracion': <django.f
ConfiguracionForm.base_fields = {'tipo_identificacion': <django.forms.fields.TypedChoiceField object at 0x2b7851d43a50>,
ConfiguracionForm.declared_fields = {}
ConfiguracionForm.media

class profiles.forms.GanaderiaForm (data=None, files=None, auto_id=u'id_%s', pre-
fix=None, initial=None, error_class=<class 'django.forms.util.ErrorList'>, label_suffix=None, empty_permitted=False, instance=None)
Clases base: django.forms.models.ModelForm

class Meta

    fields = ('nombreEntidad', 'direccion')

    model
        alias de Ganaderia

    widgets = {'nombreEntidad': <django.forms.widgets.TextInput object at 0x2b7851dc7c90>, 'direccion': <django.f
GanaderiaForm.base_fields = {'nombreEntidad': <django.forms.fields.CharField object at 0x2b7851dd5590>, 'direccion': <django.f
GanaderiaForm.declared_fields = {}
GanaderiaForm.media

class profiles.forms.SignupFormExtra (*args, **kw)
Clases base: userena.forms.SignupForm

A form to SIDGVnstrate how to add extra fields to the signup form, in this case adding the first and last name.

base_fields = {'username': <django.forms.fields.RegexField object at 0x2b7851aae190>, 'email': <django.forms.fields.EmailField object at 0x2b7851aae190>}

media

save ()
    Override the save method to save the first and last name to the user field.

```

### 3.7.4 profiles.models module

```

class profiles.models.Configuracion (*args, **kwargs)
    Clases base: django.db.models.base.Model

    Configuracion(id, tipo_identificacion, celo_frecuencia, celo_frecuencia_error, celo_duracion, celo_duracion_error, celo_despues_parto, celo_despues_parto_error, intentos_verificacion_celo, etapa_tenera, etapa_vacona, etapa_ventre, periodo_gestacion, periodo_seco, periodo_lactancia, periodo_vacio, numero_ordenios)

    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist

    Configuracion.IDENTIFICACION_CHOICES = (('simple', 'Simple'), ('norma_ecuador', 'Norma Ecuador'))

    exception Configuracion.MultipleObjectsReturned
        Clases base: django.core.exceptions.MultipleObjectsReturned

    Configuracion.ganaderia

```



```

Configuracion.get_tipo_identificacion_display(*moreargs, **morekwargs)
Configuracion.objects = <django.db.models.manager.Manager object at 0x2b7851715dd0>
class profiles.models.Ganaderia(*args, **kwargs)
    Clases base: django.db.models.base.Model
    Ganaderia(id, nombreEntidad, direccion, configuracion_id)
    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist
    exception Ganaderia.MultipleObjectsReturned
        Clases base: django.core.exceptions.MultipleObjectsReturned
    Ganaderia.configuracion
    Ganaderia.farm_foods
    Ganaderia.ganados
    Ganaderia.insemination_farm
    Ganaderia.medicaments
    Ganaderia.objects = <django.db.models.manager.Manager object at 0x2b785172d510>
    Ganaderia.perfil
class profiles.models.Profile(*args, **kwargs)
    Clases base: userena.models.UserenaLanguageBaseProfile
    Default profile
    exception DoesNotExist
        Clases base: django.core.exceptions.ObjectDoesNotExist
    Profile.GENDER_CHOICES = ((1, <django.utils.functional.__proxy__ object at 0x2b7851715d50>), (2, <django.utils.fun
    exception Profile.MultipleObjectsReturned
        Clases base: django.core.exceptions.MultipleObjectsReturned
    Profile.ganaderia_perfil
    Profile.get_gender_display(*moreargs, **morekwargs)
    Profile.get_language_display(*moreargs, **morekwargs)
    Profile.get_privacy_display(*moreargs, **morekwargs)
    Profile.objects = <userena.managers.UserenaBaseProfileManager object at 0x2b7851722710>
    Profile.receiver_
    Profile.sender_
    Profile.user

```

### 3.7.5 profiles.tests module

### 3.7.6 profiles.views module

```

profiles.views.agrega_ganaderia_config(request)
profiles.views.home(request)

```

```
profiles.views.number_messages(request, *args, **kwargs)
```

### 3.7.7 Module contents

## 3.8 Módulo de Reportes

### 3.8.1 Submodules

### 3.8.2 reports.admin module

### 3.8.3 reports.models module

### 3.8.4 reports.tests module

### 3.8.5 reports.views module

Aquí tenemos las vistas del módulo de los reportes, que son eficaces para realizar los procesos de: listar reportes y la vista en sí del reporte de ganados hembras de la ganadería. Example usage:

```
>>> @login_required
>>> def lista_ganado_produccion(request, username):
>>> id_user = User.objects.filter(username=username)
>>> try:
...     ganaderia = Ganaderia.objects.get(perfil=id_user)
>>> except ObjectDoesNotExist:
...     return redirect(reverse('agrega_ganaderia_config'))
>>> configuracion = Configuracion.objects.get(id=ganaderia.configuracion_id)

>>> if configuracion.tipo_identificacion == 'simple':
...     gg = Ganado.objects.filter(ganaderia_id=ganaderia.id, genero=1, etapas__nombre=2, ciclos__no
>>> else:
...     gg = Ganado.objects.filter(ganaderia_id=ganaderia.id, genero=1)
>>> return render_to_response('lista_ganado_produccion.html',
>>>     {'ganado':gg},
>>>     context_instance=RequestContext(request))
>>> from datetime import timedelta
>>> year = timedelta(days=365)
>>> another_year = timedelta(weeks=40, days=84, hours=23,
...                           minutes=50, seconds=600) # adds up to 365 days

>>> year.total_seconds()
31536000.0
>>> year == another_year
True
>>> ten_years = 10 * year
>>> ten_years, ten_years.days // 365
(datetime.timedelta(3650), 10)
>>> nine_years = ten_years - year
>>> nine_years, nine_years.days // 365
(datetime.timedelta(3285), 9)
>>> three_years = nine_years // 3;
>>> three_years, three_years.days // 365
(datetime.timedelta(1095), 3)
```

```
>>> abs(three_years - ten_years) == 2 * three_years + year
True
```

```
reports.views.docs(request)
```

```
reports.views.list_reports(request)
```

```
reports.views.view_report_female(request, id_cattle)
```

### 3.8.6 Module contents

## 3.9 Módulo Asíncrono

### 3.9.1 Subpackages

#### Módulo Asíncrono

##### Submodules

`webServices.wsGanados.admin` module

`webServices.wsGanados.models` module

`webServices.wsGanados.tests` module

`webServices.wsGanados.urls` module

`webServices.wsGanados.views` module

```
class webServices.wsGanados.views.AgentAlimentacion(agentjid, password, resource='spade', port=5222, debug=[], p2p=False)
```

Clases base: `spade.Agent.Agent`

```
class BehaviourAlimentacion
```

Clases base: `spade.Behaviour.OneShotBehaviour`

```
onEnd()
```

```
onStart()
```

```
class webServices.wsGanados.views.AgentProduccion(agentjid, password, resource='spade', port=5222, debug=[], p2p=False)
```

Clases base: `spade.Agent.Agent`

```
class BehaviourProduccion
```

Clases base: `spade.Behaviour.OneShotBehaviour`

```
onEnd()
```

```
onStart()
```

```
class webServices.wsGanados.views.AgentReproduccion(agentjid, password, resource='spade', port=5222, debug=[], p2p=False)
```

Clases base: `spade.Agent.Agent`

```

class BehaviourReproduccion
    Clases base: spade.Behaviour.OneShotBehaviour
    onEnd()
    onStart()

class webServices.wsGanados.views.AgentSanidad(agentjid, password, resource='spade',
                                                port=5222, debug=[], p2p=False)
    Clases base: spade.Agent.Agent
    class BehaviourSanidad
        Clases base: spade.Behaviour.OneShotBehaviour
        onEnd()
        onStart()

class webServices.wsGanados.views.BeliefAlimentacion
class webServices.wsGanados.views.BeliefProduccion
class webServices.wsGanados.views.BeliefReproduccion
class webServices.wsGanados.views.BeliefSanidad
class webServices.wsGanados.views.DesireAlimentacion
class webServices.wsGanados.views.DesireProduccion
class webServices.wsGanados.views.DesireReproduccion
class webServices.wsGanados.views.DesireSanidad
class webServices.wsGanados.views.IntentionAlimentacion

    assignNotification(d, datee, name)
    sendNotificationAmountFood(desire_amount_food)
    sendNotificationExpirationFood(desire_expiration_food)

class webServices.wsGanados.views.IntentionProduction

    assignNotification(d)
    sendNotificationOrdenio(desire_ordenio)

class webServices.wsGanados.views.IntentionReproduccion

    assignNotification(d, datee, name)
    sendNotificationCelo(desire_celo)
    sendNotificationPajuela(desire_pajuelas)
    sendNotificationParto(desire_parto)
    sendNotificationService(desire_service)
    sendNotificationVerification(desire_verification)

class webServices.wsGanados.views.IntentionSanidad

    assignNotification(d, datee, name)

```

```

sendNotificationAmountVaccine (desire_amount_vaccine)
sendNotificationAmountWormer (desire_amount_wormer)
sendNotificationExpirationVaccine (desire_expiration_vaccine)
sendNotificationExpirationWormer (desire_expiration_wormer)

webServices.wsGanados.views.ajaxAssignCattleFoodFinal (request, *args, **kwargs)
webServices.wsGanados.views.ajaxAssignCattleFood_view (request, *args, **kwargs)
webServices.wsGanados.views.ajaxAssignCattleVaccineFinal (request, *args, **kwargs)
webServices.wsGanados.views.ajaxAssignCattleVaccine_view (request, *args, **kwargs)
webServices.wsGanados.views.ajaxAssignCattleWormerFinal (request, *args, **kwargs)
webServices.wsGanados.views.ajaxAssignCattleWormer_view (request, *args, **kwargs)
webServices.wsGanados.views.ajaxCattleMaleRp_view (request, *args, **kwargs)
webServices.wsGanados.views.ajaxCattleMale_view (request, *args, **kwargs)
webServices.wsGanados.views.ajaxFood_view (request, *args, **kwargs)
webServices.wsGanados.views.ajaxGanadosMachosInseminacion_view (request, *args, **kwargs)
webServices.wsGanados.views.ajaxInsemination_view (request, *args, **kwargs)
webServices.wsGanados.views.ajaxRefresh (request, *args, **kwargs)
webServices.wsGanados.views.ajaxVaccine_view (request, *args, **kwargs)
webServices.wsGanados.views.wsGanadosHembras_view (request, *args, **kwargs)
webServices.wsGanados.views.wsGanadosProduccion_view (request, *args, **kwargs)
webServices.wsGanados.views.wsGanados_view (request, *args, **kwargs)
webServices.wsGanados.views.wsWormer_view (request, *args, **kwargs)

```

## Module contents

### 3.9.2 Module contents



---

## Índice y Tablas

---

- *genindex*
- *modindex*
- *search*





## a

alimentos, 12  
alimentos.admin, 10  
alimentos.forms, 10  
alimentos.models, 10  
alimentos.tests, 11  
alimentos.views, 11

## g

ganados, 22  
ganados.admin, 12  
ganados.forms, 12  
ganados.models, 16  
ganados.tests, 21  
ganados.views, 21

## m

medicament, 25  
medicament.admin, 22  
medicament.forms, 22  
medicament.models, 23  
medicament.tests, 24  
medicament.views, 24  
messages, 26  
messages.admin, 25  
messages.forms, 25  
messages.models, 25  
messages.tests, 26  
messages.views, 26

## n

notifications, 27  
notifications.admin, 26  
notifications.models, 26  
notifications.tests, 27  
notifications.views, 27

## p

profiles, 30  
profiles.admin, 27

profiles.forms, 27  
profiles.models, 28  
profiles.tests, 29  
profiles.views, 29

## r

reports, 31  
reports.admin, 30  
reports.models, 30  
reports.tests, 30  
reports.views, 31

## s

SIDGV, 9  
SIDGV.settings, 9  
SIDGV.urls, 9  
SIDGV.wsgi, 9

## w

webServices, 33  
webServices.wsGanados, 33  
webServices.wsGanados.admin, 31  
webServices.wsGanados.models, 31  
webServices.wsGanados.tests, 31  
webServices.wsGanados.urls, 31  
webServices.wsGanados.views, 31