
e-cidadania

herramienta de e-democracia para la ciudadanía

Documentacion

Release

Cidadania S. Coop. Galega

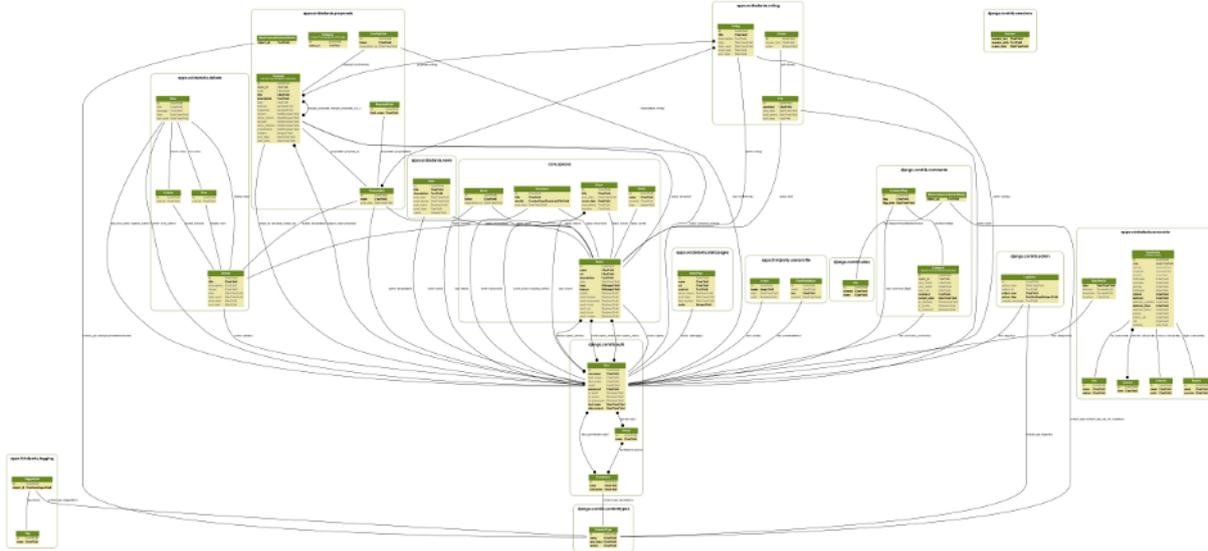
December 07, 2013

Contents

1	Instalación y configuración	3
1.1	Instalación	3
1.2	Configuración	4
1.3	Despliegue	6
2	Manuales	7
2.1	Manual de usuario	7
2.2	Manual de administración	9
3	Apariencia / Temas	13
3.1	Apariencias de e-ciudadania	13
4	Desarrollo	15
4.1	Guía de estilo	15
4.2	Cuentas de usuario	17
4.3	Creando módulos	18
4.4	Generando la documentación	18
4.5	Traducciones	19
5	Referencia (inglés)	23
5.1	apps.ecidadania.cal — Calendar and events	23
5.2	apps.ecidadania.debate — Debates	24
5.3	core.spaces — Spaces/Processes	24
5.4	apps.ecidadania.proposals — Proposals	25
5.5	apps.ecidadania.news — News	25
6	Colofón	27
6.1	Conseguir ayuda	27
	Python Module Index	29

e-cidadania es una plataforma de e-democracia hecha con software abierto para la participación ciudadana. Las características clave son: un sistema de propuestas, un revolucionario sistema de debate ordenado, repositorio de documentos, informes autogenerados y perfiles avanzados de usuario entre otras cosas como geolocalización y mensajería.

Esta plataforma está basada en el framework [django](#) y algunas librerías externas.



Warning: e-cidadania está en desarrollo, y por ello algunos componentes pueden variar sus características, especialmente los modelos de datos y esta documentación.

Instalación y configuración

1.1 Instalación

La instalación de e-cidadania es muy simple y se realiza de la misma manera que cualquier otra plataforma django estándar.

1.1.1 Descargar la plataforma

Note: La sección de descargas de la web oficial todavía no está disponible.

Existen numerosas formas de descargar e-cidadania. La más sencilla de todas es ir a la sección [descargas](#) de la web y bajarse la última versión estable o de desarrollo, listas para utilizar.

Versión estable

Puedes encontrar la última versión estable en la página de descargas de e-cidadania.org:

```
http://ecidadania.org/downloads
```

Versión de desarrollo

La versión de desarrollo está disponible desde varios sitios. Utilizamos [GIT](#) como sistema de control de versiones, así que tendrás que instalarlo en tu ordenador.

Gitorious: (*repositorio oficial*):

```
git clone git://gitorious.org/e-cidadania/mainline.git
```

GitHub (*repositorio secundario*):

```
git clone git://github.com/oscarcp/e-cidadania.git
```

Repo.or.cz (*espejo oficial*):

```
git clone git://repo.or.cz/e_cidadania.git
```

1.1.2 Requisitos

- Python 2.5 o superior
- Apache, nginx, u otro servidor web con soporte CGI
- FastCGI, CGI, Passenger u otro CGI

Dependencias

- django 1.3
- PIL (*Python Imaging Library*)
- python-datetime (*versión 1.5*)
- django-tagging
- django-wysiwyg
- django-grappelli (para la administración)
- feedparser

1.1.3 Instalar dependencias

Puedes instalar todas las dependencias automáticamente con este comando:

```
# pip install -r requirements.txt
```

Si no dispones del programa *pip* también puedes instalar las dependencias a través del programa *easy_install*.

1.2 Configuración

La plataforma e-cidadania viene casi lista para funcionar tan pronto la descargues, pero aún te queda editar el archivo *settings.py* y algunos pasos más.

1.2.1 Base de datos (BDD)

Configurando la base de datos:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': 'db/sqlite.db',  
        'USER': '',  
        'PASSWORD': '',  
        'HOST': '',  
        'PORT': '',  
    }  
}
```

Lo primero de todo es configurar la base de datos. Por defecto e-cidadania viene configurado para utilizar una base de datos local de tipo SQLite 3, que te puede resultar útil para realizar pruebas antes de poner la plataforma en producción, pero que deberás cambiar tan pronto como termines las pruebas.

Un ejemplo de base de datos en un servidor compartido de DreamHost es este:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'ecidadania_database',  
        'USER': 'databaseadmin',
```

```
        'PASSWORD' : 'somepassword',  
        'HOST' : 'mysql.ecidadania.org',  
        'PORT' : '',  
    }  
}
```

1.2.2 Modo Debug

El modo debug viene activado por defecto y se recomienda encarecidamente desactivarlo para comenzar a utilizar e-cidadania en producción. Para ello hay que desactivarlo en el fichero *settings.py*:

```
DEBUG = False
```

1.2.3 Perfiles de usuario

ACCOUNT_ACTIVATION_DAYS (número)

Esta variable especifica cuántos días tiene el usuario para activar su cuenta desde que recibe el correo de confirmación.

GOOGLE_MAPS_API_KEY (hash)

Llave de la API de Google para poder utilizar la interfaz de mapas. Debes crearte una propia a pesar de que e-cidadania venga con una configurada, ya que sólo funcionará en el dominio que hayas especificado.

1.2.4 Correo electrónico

ADMINS (lista)

Lista de administradores y cuentas de correo para la notificación de errores del servidor. Sólo funciona si `DEBUG = False`

EMAIL_HOST (servidor)

Servidor de correo desde el cual se enviarán los correos a los usuarios.

DEFAULT_FROM_EMAIL

Dirección por defecto desde la que se enviarán los correos si no se especifica otra.

1.2.5 Idioma

LANGUAGE_CODE (código de idioma)

Idioma con el cual funcionará django por defecto.

1.2.6 Después de settings.py

Después de configurar e-cidadania a tu gusto, tendrás que ejecutar una serie de comandos para que todo esté en orden.

Crear la BDD

Para crear la base de datos con el primer usuario de administración ejecutamos desde la raíz del proyecto:

```
./manage.py syncdb
```

Crearé las tablas de la base de datos y posteriormente nos preguntará si queremos crear un usuario de administración. Elegiremos la opción que más nos convenga y seguiremos.

En principio con esto es suficiente. Si por algún motivo quieres meter un set de datos previos, deberás hacerlo a través de los métodos que Django ofrece, pero eso cae fuera de este manual.

Recoger ficheros estáticos

e-cidadania viene configurado para servir los ficheros estáticos tanto en desarrollo como en producción, pero en este último hay que *recolectarlos* y juntarlos en un directorio.

Ese directorio viene predeterminado como “static”, y los ficheros estáticos propios de e-cidadania están almacenados en “static_files”. Para *recolectar* los ficheros debes ejecutar el comando:

```
./manage.py collectstatic
```

Tras ejecutarlo puedes borrar el directorio *static_files* si quieres, aunque recomendamos mantenerlo por si algún día necesitas ejecutar el servidor de desarrollo.

1.2.7 Plugins

Note: e-cidadania aún no soporta plugins, pero lo hará en un futuro.

1.3 Despliegue

1.3.1 Apache 2

Esta sección está en desarrollo.

1.3.2 nginx + FastCGI

Note: La instalación a través de FastCGI debería funcionar para otros tipos de servidor, ya que FastCGI es el que se encarga de ejecutar la aplicación mientras que el servidor sólo se encarga del contenido estático.

Warning: Esta sección está en desarrollo.
--

1.3.3 DreamHost

Las instrucciones de instalación para un servidor compartido en DreamHost son demasiado extensas para incluir en esta documentación, por favor, visita el blog de [Oscar Carballal](#) para ver cómo hacerlo.

Manuales

2.1 Manual de usuario

Bienvenido al manual de usuario de e-cidadanía. Este manual está destinado a todos los nuevos usuarios de la plataforma. El objetivo es cubrir lo mejor posible todas las funciones de e-cidadanía evitando ser demasiado específicos.

2.1.1 Premisas

- e-cidadanía será siempre libre y de código abierto.
- Siempre daremos las mejores traducciones y opciones de accesibilidad posibles.
- Bajo ningún concepto se registrará la actividad de los usuarios que no sea relevante para el proceso.

2.1.2 Cómo funciona e-cidadanía

e-cidadanía funciona de una forma muy similar a un gestor de proyectos. Por una parte tenemos la página principal, en la que podremos ver las noticias más relevantes relacionadas con los procesos en curso, así como opciones para las redes sociales o elegir el espacio al que queremos ir.

La filosofía de los espacios es muy simple. Cada proceso de participación es un espacio, que puede ser público o privado. Dentro de cada espacio hay una serie de herramientas que el administrador del mismo podrá configurar a su gusto como por ejemplo: noticias, repositorio de documentos, sistema de propuestas, sistema de debates, seguimiento de reuniones, etc.

No se puede acceder a los procesos salvo que se haya sido invitado, o que el proceso sea público. En cualquier caso, para participar en un proceso hay que ser admitido en el mismo.

2.1.3 Registrarse en la plataforma

El sistema de registro de usuarios dependerá de lo que el administrador quiera. Por defecto, e-cidadanía permite el registro de usuarios para poder realizar algunas tareas básicas, y visitar los espacios públicos, pero sin poder participar.

Para registrarte debes hacer clic en “Entrar” y en la parte inferior del formulario de entrada te aparecerá la opción de registro. Basta con tu nombre, correo electrónico y clave.

Recuerda que esto sólo te permitirá ver los espacios públicos pero no podrás participar en ninguno salvo que seas incluido en él.

2.1.4 Cómo participar en un espacio

Los usuarios nuevos que se hayan registrado automáticamente no tienen acceso a ningún espacio de la plataforma salvo los públicos, que podrán ver pero no participar en ellos.

Para poder participar en un espacio es necesario solicitar el acceso. Eso se puede realizar o bien enviando un correo electrónico al administrador responsable de ese espacio, o entrando en el espacio y haciendo clic en el botón “Quiero participar”, tras lo cual será enviada una notificación al administrador del espacio para que evalúe tu participación.

Otro aspecto importante de estar en un proceso de participación es que, dependiendo del tipo de proceso puede que necesites rellenar tu perfil de usuario, que incluye datos personales.

Para los procesos oficiales esto seguramente sea una obligación, pero no para los procesos que inician los usuarios por su cuenta.

2.1.5 Propuestas

El módulo de propuestas permite realizar propuestas para que la gente las vote o las apoye.

Ese tipo de valoraciones se tendrán después en cuenta en el proceso de votación definitivo.

Enviar una propuesta es de lo más sencillo. Basta con acceder al espacio de participación correspondiente y hacer clic en “Añadir una propuesta” en la columna de la derecha.

Aparecerá un formulario que tiene varios campos:

Título El nombre de la propuesta. Debe ser una síntesis concisa de la propuesta que no debe ocupar demasiado.

Descripción Una descripción más extensa de la propuesta. Se pueden vincular elementos externos como páginas web, imágenes, vídeos, etc.

Latitud/Longitud Ahora mismo e-ciudadania cuenta con un sistema básico de geoposicionamiento. Si quieres situar tu propuesta en un mapa tendrás que poner las coordenadas.

Etiquetas Las etiquetas identifican de forma sencilla tu propuesta, por ejemplo:

```
Titulo: La acera de Cobián Roffignac está rota
Etiquetas: acera, cobian roffignac, pontevedra, roto
```

Warning: Una vez hecha una propuesta no se podrá editar ni eliminar. Este comportamiento es susceptible de cambiar en e-ciudadania 0.1.5

2.1.6 Debates

El sistema de debates es nuevo, así que procura prestar atención para no perderte, esto no es un foro al uso.

Se han trasladado diversos modelos de debate presencial hasta esta plataforma, y uno de ellos es el que te vamos a enseñar ahora. ¡Ya verás qué rápido te acostumbras!

El sistema de debate consta de un *tablón* en donde se ponen las notas con las ideas de cada uno. Las notas están limitadas, para evitar una sobrecarga del tablón.

Para añadir tus ideas, basta con que hagas clic en el botón “Nueva nota” y se creará una nueva nota, en la que podrás escribir. Después basta con que la arrastres a la zona del tablón que quieras.

2.1.7 Cómo consultar documentos

Basta con hacer clic en el documento que quieras leer para que comience a descargarse. Si el documento que quieres no figura en la columna de la derecha puedes darle al botón “Ver todos los documentos” y se cargará una página nueva que te mostrará todos los documentos que están almacenados en ese espacio.

2.1.8 Suscripción RSS

Los espacios, así como la página principal, cuentan con feeds RSS para poder estar al día con tu lector de noticias favorito en vez de tener que visitar la página web constantemente.

Para obtener el enlace del feed, basta con que hagas clic abajo de todo en “RSS Feed”.

Ten en cuenta que cada espacio tiene un feed diferente, así como la página principal de e-ciudadania. Si quieres un feed sólo de tu espacio deberás hacer clic en el icono RSS de tu espacio y no en el de la página principal.

Note: En e-ciudadania 0.1.5 se espera tener un único botón de RSS que llevará al usuario a un selector de feeds RSS.

2.1.9 Preguntas frecuentes

- ¿Puedo crear yo mi propio espacio de participación?

Si, pero actualmente hay que solicitarlo al administrador de la plataforma ya que e-ciudadania no permite la creación de espacios a nadie salvo a los administradores.

Note: Esto es susceptible de cambiar en un futuro.

- ¿Puedo participar en un espacio privado?

Depende de las condiciones que se hayan establecido para el espacio. En el caso de los presupuestos participativos las condiciones son muy estrictas y no podrás participar a menos que las reúnas.

- ¿Existe aplicación para teléfonos móviles?

e-ciudadania ha sido diseñado y programado para que la interfaz funcione correctamente en los smartphones actuales, no hay necesidad de una aplicación aparte. Basta con que visites la web normalmente :)

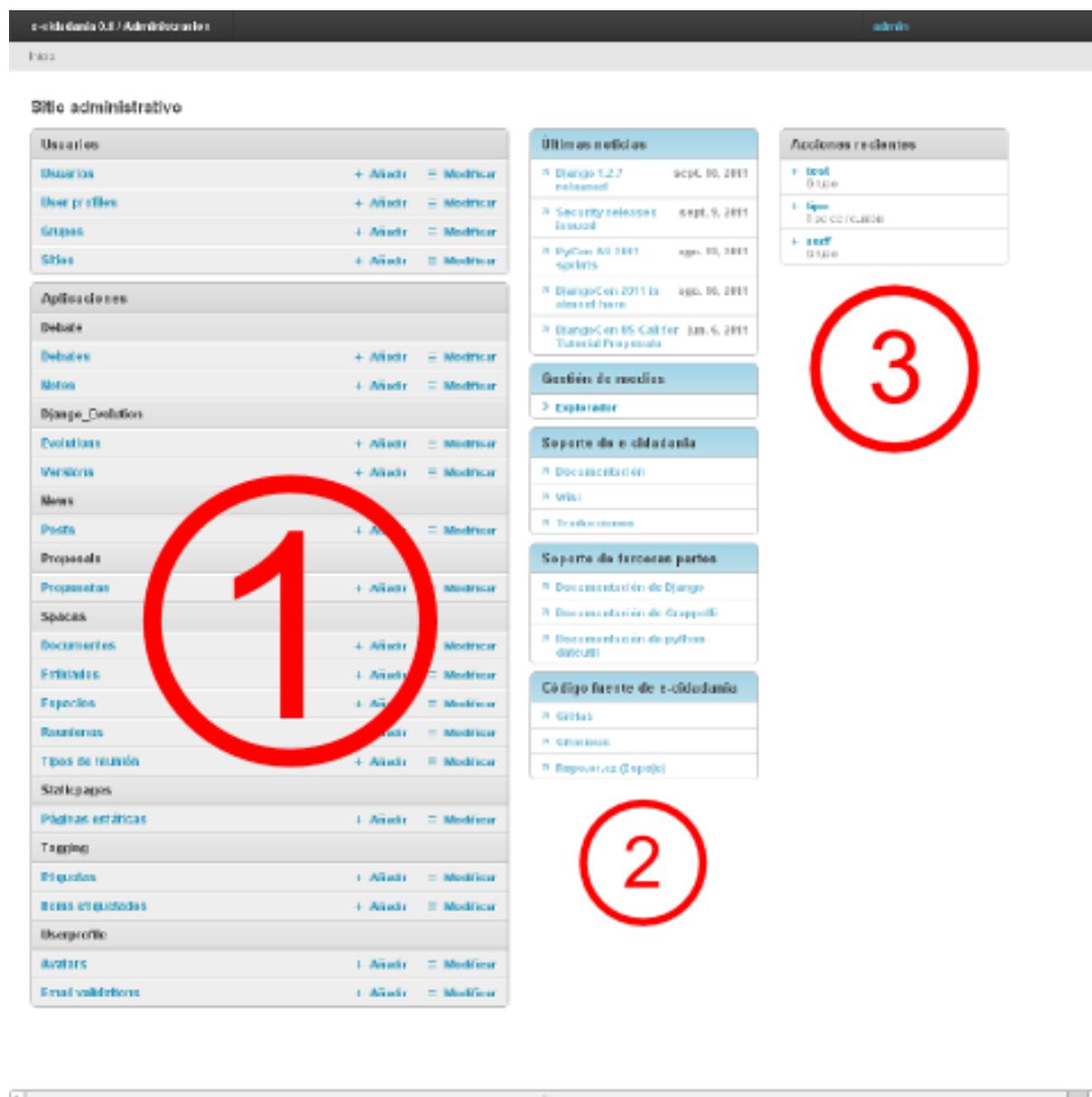
Si tienes alguna pregunta que no está aquí por favor dínoslo!

2.2 Manual de administración

Este manual te enseñará cómo manejar e-ciudadania para que no te pierdas nada de nada :).

Concretamente trata la parte del administrador del sitio sin meterse en detalles técnicos innecesarios. Si en algún momento necesitas detalles técnicos es preferible que leas el manual de desarrollo, en el cual está todo especificado.

2.2.1 Estructura del panel de administración



La página principal del panel de administración se divide en tres grandes zonas:

1. Esta zona contiene todas las aplicaciones de la plataforma. Desde aquí podrás añadir, editar o eliminar cualquier elemento de la base de datos.

Note: Puede que aparezcan también otras aplicaciones de terceros, como `django-evolution` o similares, que son para la gestión de ciertos aspectos técnicos de la plataforma.

2. Esta zona contiene lo que llamamos *widjets*. En ellos se puede mostrar casi cualquier tipo de información. Normalmente esos widjets vienen configurados por defecto. Si quieres cambiar su contenido basta con que edites el archivo `dashboard.py` que hay en la raíz de la aplicación, pero ojo, no te lo recomendamos si no sabes lo que estás haciendo. Si aún así quieres hacerlo, mira el [manual de django-grappelli](#).
3. Esta es la zona de registro. Aquí se mostrará toda la actividad reciente que haya ocurrido en la plataforma.

2.2.2 Registro de usuarios

Los usuarios tienen dos formas de registrarse:

- **A través del proceso participativo presencial** Se anotan para participar en el proceso, y se vuelcan sus datos en la base de datos de e-ciudadanía.

En este caso tendrás que crear su usuario desde “*Usuarios*” y luego el perfil correspondiente desde “*User profiles*”

- **Registrándose ellos mismos** Esta es la forma más rápida pero también menos conveniente. Con este registro el usuario no tendrá ninguno de sus datos personales en la base de datos, ni pertenecerá a ningún proceso participativo salvo que se le permita. De la misma forma, sólo podrá visualizar los procesos participativos abiertos, pero no podrá participar a menos que se le incluya en el grupo.

Note: Este comportamiento es susceptible de cambiar en un futuro.

Por defecto e-ciudadanía tiene el registro de usuarios activado cuando se instala. Para desactivarlo tendrás que comentar una línea de código en el siguiente archivo:

```
apps/userprofile/urls/en.py:107
    # url(r'^register/$', register, name='signup'),
```

2.2.3 Permisos

Note: En e-ciudadanía 0.2 se actualizará a permisos por fila, que permiten un nivel mucho más detallado de permisos.

Los permisos en e-ciudadanía están heredados directamente del sistema de autenticación de django. De esa forma ya disponemos de permisos basados en grupo o individuales.

- **Por grupo** Lo ideal es gestionar los usuarios a través de grupos. Se puede hacer accediendo al panel de administración desde el menú que hay arriba a la derecha, al lado del nombre de usuario.

Una vez en el panel de administración basta con que hagamos clic en el botón que hay al lado de “*Grupos*” que pone “ + *Añadir*” y ahí tendremos opción a configurar nuestros grupos.

Como podrás observar, tenemos dos paneles de permisos y un campo para escribir el nombre. Basta con que pases los permisos que quieras para ese grupo al panel de la derecha (mediante las flechas que hay entre los dos paneles) y luego le pongas un nombre al grupo.

Para guardar haz clic abajo a la derecha donde pone “*Guardar*”.

- **Individuales** Puede que en algún momento quieras que alguien tenga una serie de permisos en concreto que no encajan en ninguno de los grupos que has creado. En ese caso es mejor darle los permisos al usuario en sí en vez de crear un grupo para él solo.

Para ello debes editar al usuario haciendo clic en “*Usuarios*” y seleccionando el que quieras editar. Se te mostrarán una serie de datos del usuario como los grupos a los que pertenece, fecha de registro, etc. Lo que nos interesa son los paneles de permisos.

Funcionan exactamente igual que los de grupo. Basta con que lo configures y guardes los cambios para aplicar los permisos al usuario.

2.2.4 Espacios

Los espacios son zonas separadas en las cuales tienen lugar los procesos participativos. Cada espacio puede configurarse a medida de las necesidades del proceso y no podrá participar ningún usuario que no haya sido “admitido”.

La creación de espacios corresponde al administrador de la plataforma, por medidas de seguridad.

Note: Esta sección está sin terminar.

2.2.5 Módulos

e-ciudadania es una plataforma totalmente modular, incluso las características más básicas como noticias, repositorio de documentos, espacios, sistema de propuestas, etc. son módulos que pueden ser modificados o reemplazados sin afectar a la estructura de la plataforma.

2.2.6 Moderación

Las tareas de moderación dentro de la plataforma son muy sencillas. Cada módulo o aplicación tiene tres tareas básicas: crear, editar y eliminar.

Crear Dependiendo del grado de moderación que tengas, podrás añadir contenido de distintos tipos. Los niveles de moderación más altos permiten crear prácticamente cualquier cosa y con un elevado grado de detalle.

Editar La tarea de edición es similar a la de creación, con la diferencia de que se te devolverá un formulario con los datos del objeto que estés editando.

El contenido generado por los usuarios debe ser preservado, sólo puede ser eliminado por los administradores del sitio.

Eliminar Normalmente en un foro, un moderador puede eliminar el contenido de un usuario. En e-ciudadania ese no es el objetivo. Todo el contenido generado por los usuarios debe preservarse y sólo podrá ser eliminado por los administradores del sitio o bajo petición del propio usuario.

2.2.7 Errores frecuentes

Lista de errores más frecuentes a la hora de administrar.

Apariencia / Temas

3.1 Apariencias de e-cidadania

3.1.1 Otras apariencias

Este es un pequeño índice de apariencias de e-cidadania.

3.1.2 Crea tu propia apariencia para e-cidadania

Crear una nueva apariencia para e-cidadania puede llegar a ser algo difícil, pero no por ello imposible. Aquí te daremos las instrucciones precisas.

3.1.3 Referencia

Los temas de e-cidadania se localizan mayoritariamente en los propios módulos. Cada aplicación gestiona por separado sus apariencias, no en cuanto a estilo pero sí en cuanto a distribución.

Además de esto, existen algunas platillas generales que están situadas en el directorio *templates*.

General

asdasd

Perfil de usuario

asdasdas

Propuestas

sdfsdfs

Debates

sdfsdfs

Noticias

sdfsdf

Documentos

Desarrollo

4.1 Guía de estilo

La guía de estilo establece una serie de normas a seguir cuando se programa en e-cidadania. Estas normas son *inquebrantables*. La guía de estilo sigue muy de cerca el documento [PEP8](#), con algunas excepciones que vienen de la guía de estilo interna de [Pocoo](#).

4.1.1 Python

Imports Todos los imports deben estar situados en la cabecera del archivo, por debajo del comentario de cabecera. Los imports de módulos python deben preceder a todos los demás, y las librerías externas o módulos de terceras partes deben preceder a los de las aplicacion.

Ejemplo:

```
import os
import sys

from extlib import function

from myapp.module import function
```

Ancho de línea (columnas) El código debe ser siempre de 80 columnas de ancho, se permite un par de columnas extra en casos de necesidad.

Declaraciones largas Si una línea de código no cabe en 80 columnas, intenta reducirlo declarando variables previamente. Si todavía no encaja, puedes dividir las líneas de la siguiente forma:

Sentencias con paréntesis:

```
website = models.URLField(_('Website'), verify_exists=True,
                           max_length=200, null=True, blank=True,
                           help_text=_('The URL will be checked'))
```

Declaraciones:

```
this_is_a_very_long(function_call, 'with many parameters') \
    .that_returns_an_object_with_an_attribute

MyModel.query.filter(MyModel.scalar > 120) \
    .order_by(MyModel.name.desc()) \
    .limit(10)
```

Listas, tuplas y diccionarios:

```
items = [  
    'this is the first', 'set of items', 'with more items',  
    'to come in this line', 'like this'  
]  
  
dict = {  
    ('mobile': phone),  
    ('car': key),  
    ('another': thing),  
}
```

Indentación La indentación debe ser **siempre** de 4 espacios por nivel. No se permite el uso de tabulaciones para indentar.

Líneas en blanco Cada función y clase debe de estar separado por dos líneas en blanco. El código dentro de una clase o método debe de estar separado por una línea en blanco.

Ejemplo:

```
class ListDocs(ListView):  
    ----blank line----  
    """  
    List all documents stored within a space.  
    """  
    paginate_by = 25  
    context_object_name = 'document_list'  
    ----blank line----  
    def get_queryset(self):  
        place = get_object_or_404(Space, url=self.kwargs['space_name'])  
        objects = Document.objects.all().filter(space=place.id).order_by('pub_date')  
        return objects  
    ----blank line----  
    def get_context_data(self, **kwargs):  
        context = super(ListDocs, self).get_context_data(**kwargs)  
        context['get_place'] = get_object_or_404(Space, url=self.kwargs['space_name'])  
        return context  
    ----blank line----  
    ----blank line----  
    def whatever(args):  
        ----blank line----  
        """  
        A comment.  
        """  
        this_is_something = 0
```

4.1.2 HTML

Columnas El código HTML no tiene límite de columnas, pero debe estar indentado de forma que se pueda localizar rápidamente cada elemento en el documento. El estilo de indentación precede al resultado renderizado en el navegador.

Indentación El código X/HTML debe de estar indentado con 4 espacios por nivel igual que el código python, sin excepciones.

Note: Puede que haya código antiguo que siga la anterior norma de indentación que establecía dos espacios por nivel. Si ves algún archivo así agradeceríamos que nos enviases un parche para solucionarlo.

4.1.3 CSS

Indentación La indentación es de 4 espacios, igual que en el código python.

Ejemplo:

```
body {
  background: #FAFAFA;
  padding: 0;
  margin: 0;
  font-family: Verdana, "Lucida Sans", Arial;
  font-size: 1em;
  color: #000;
  cursor: default;
}
```

Colores Los colores deben ser escritos siempre en hexadecimal. Se permite utilizar abreviaturas de tres dígitos.

Tamaño de fuente El tamaño de fuente debe estar especificado siempre en **em's** salvo que sea un requisito de la presentación.

4.1.4 JavaScript

Al código javascript se le aplican las mismas normas que al código python.

4.2 Cuentas de usuario

El módulo de cuentas de usuario se basa en el módulo *auth* de django y en la aplicación *django-userprofile* creada por James Bennet.

Puede que para alguna instalación concreta de e-cidadania se necesiten otros datos del usuario que no tengan que ver con el modelo de datos que trae e-cidadania por defecto. En ese caso, se puede modificar los campos del perfil de usuario.

El fichero que contiene el modelo de datos del perfil se encuentra en *apps/accounts/models.py*. Todos los campos pueden ser modificados salvo los de edad y espacios.

Note: Recuerda que tras modificar el modelo de datos de tu instalación deberás reconstruir la base de datos. Recomendamos que instales alguna aplicación para migración de esquemas de bases de datos como *django-evolution*

Los usuarios constan de dos partes. Por una está la cuenta de usuario creada por el módulo *auth* de django y por otra está el objeto *profile* que contiene todos los datos del perfil.

Los usuarios pueden crearse sin su respectivo perfil, pero una vez que se intente crear un perfil habrá que rellenarlo entero y quedará asociado a su usuario de por vida.

4.2.1 Perfiles públicos

Los usuarios cuentan con una parte pública de su perfil, que mostrará los datos que ellos elijan hacer públicos.

Los datos que elijan hacerse públicos serán visibles para cualquier persona que visite el perfil. Sea usuario de la plataforma o no.

Note: Ten cuidado con la información que muestras en la red.

Warning: Actualmente los datos del perfil público no son configurables por los usuarios. Se espera cambiar esto en e-cidadania 0.1.5.

4.3 Creando módulos

e-cidadania es extensible a través de módulos, incluso antes de que se instale.

4.3.1 Estructura

Un módulo es básicamente una aplicación de django que incorporamos a e-cidadania. En estos momentos apostamos por la estructura estándar en cuanto a distribución y nombrado de archivos.

Warning: Recomendamos encarecidamente que se tenga experiencia previa en django y python antes de crear un módulo.

Un módulo consta de tres partes básicas, el modelo, la vista y la plantilla.

Note: Valorar si se debe explicar la creación de un módulo.

4.4 Generando la documentación

La documentación de e-cidadania se genera a través del programa sphinx (1.1) en tres idiomas por defecto, que son:

- Inglés
- Español
- Gallego

La estructura actual de la documentación es:

```
docs/  
  en/  
  es/  
  gl/  
  reference/  
  build/  
  dev/ *  
  docs/ *  
  theming/ *  
  index.rst *  
  authors.rst *
```

Las carpetas *en*, *es* y *gl* contienen la documentación para sus respectivos idiomas. La carpeta *reference* contiene la documentación que se genera automáticamente a partir de los comentarios de código, que serán siempre en inglés y por lo tanto estará enlazado a todos los idiomas para tener únicamente una copia y no tener que actualizar las tres de cada vez.

Lo mismo sucede para los últimos archivos y carpetas. Por motivos de compatibilidad hemos dejado la estructura “típica” de un proyecto de sphinx, que sólo son enlaces a la documentación en inglés. Esto nos permite publicar la documentación en sistios como ‘*Read The Docs* <<http://readthedocs.org>>’ que utilizan la estructura estándar para analizar la documentación.

La documentación se estructura en tres grandes categorías:

```
dev/      -- Relacionado con desarrollo o administración (instalación, configuración, puesta en ma  
docs/     -- Documentación habitual (manual de usuario, de administrador, etc.)  
theming/  -- Tutoriales sobre temas, índice de apariencias de e-cidadania, etc.
```

La documentación está estructurada de forma que sea lo más automática posible. Los tres idiomas se generan de una sentada al ejecutar *make html*.

Una vez realizado ese comando la documentación quedará distribuída de la siguiente forma:

```
build/  
  doctrees/  
  html/  
    es/  
    en/  
    gl/  
  latex/  
    es/  
    en/  
    gl/
```

Las carpetas *es*, *en* y *gl* contienen la documentación en sus respectivos idiomas. La carpeta *doctrees* sólo sirve de referencia para generar los documentos.

Como puedes ver, hay dos formatos de documentación: *latex* y *html*. e-cidadania tiene tres tipos de formato soportados oficialmente para la documentación: html, latex y pdf (generado a través de pdflatex y guardado en la carpeta *latex*).

Note: Tal como hemos diseñado el sistema multiidioma debería permitir generar cualquiera de los formatos permitidos por sphinx (como epub, manpages, htmlhelp, etc.) pero no lo garantizamos.

De la misma forma que se pueden generar todos los idiomas de una vez, también se pueden generar por separado ejecutando el mismo comando *make* pero dentro de la carpeta del idioma.

4.4.1 Normas lingüísticas

- Siempre tratar de tú al lector, nunca de usted.
- Siempre que una palabra tenga un equivalente en castellano, usar el equivalente aunque sea más común el anglicismo.

Note: Esta sección está sin terminar.

4.4.2 Palabras difusas

Note: Esta sección está en constante cambio.

Inglés	Español
Click	Clic
Delete	Eliminar
Widget	
Dashboard	
File	Archivo

4.5 Traducciones

Para las traducciones utilizamos dos herramientas:

- django-rosetta
- gettext

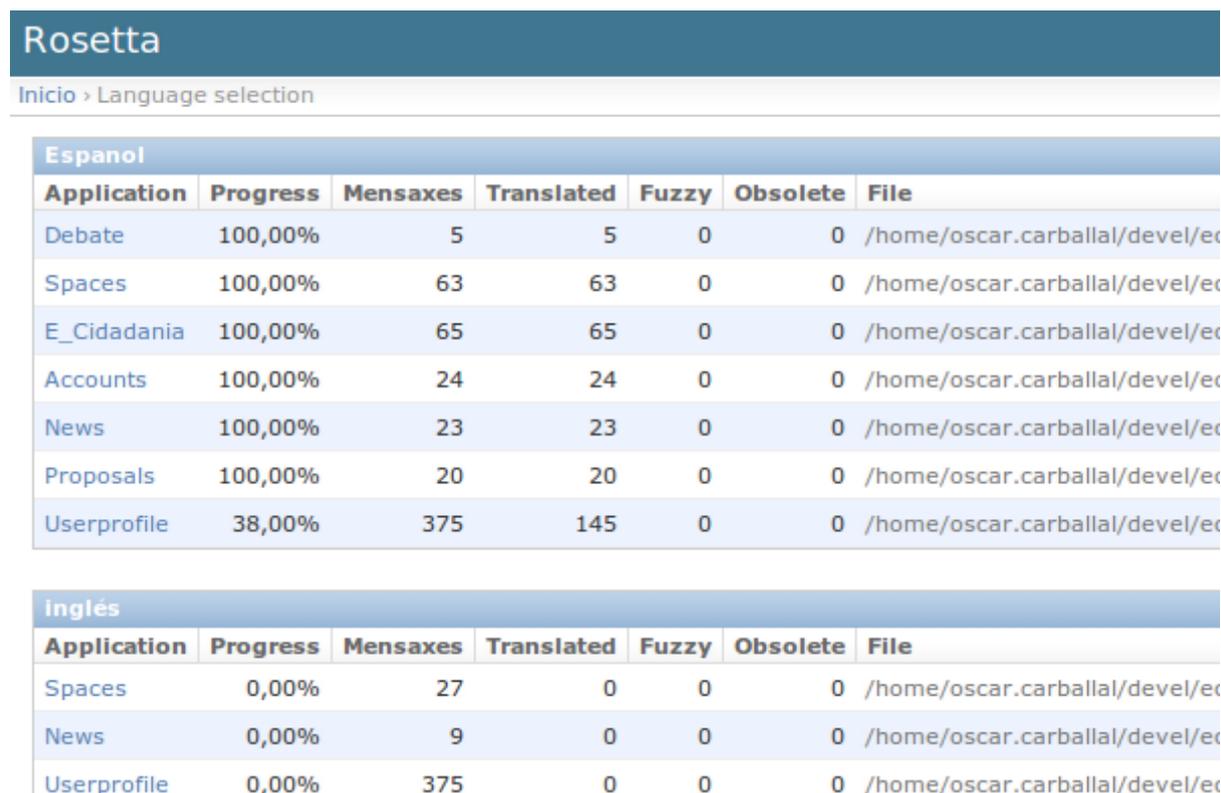
Ambas formas de traducir son sencillas gracias al *middleware* de django.

En vez de hacer un fichero de traducción global, hemos decidido mantener un fichero de traducción para cada aplicación, de esa forma aunque se cambien los módulos, las traducciones se mantendrán.

4.5.1 Traduciendo con rosetta

Para realizar traducciones con rosetta necesitarás una cuenta en el sistema y estar en el grupo “**traductores**”. Una vez hayas conseguido eso, lo demás es simple.

Tan sólo tienes que acceder a la URL de traducción y lo primero que verás será una lista con los idiomas disponibles para traducir.



The screenshot shows the Rosetta web interface. At the top, there's a header "Rosetta" and a breadcrumb "Inicio > Language selection". Below this, there are two tables. The first table is for "Español" and the second is for "Inglés". Both tables have columns for Application, Progress, Mensajes, Translated, Fuzzy, Obsolete, and File.

Español						
Application	Progress	Mensajes	Translated	Fuzzy	Obsolete	File
Debate	100,00%	5	5	0	0	/home/oscar.carballal/devel/ec
Spaces	100,00%	63	63	0	0	/home/oscar.carballal/devel/ec
E_Ciudadania	100,00%	65	65	0	0	/home/oscar.carballal/devel/ec
Accounts	100,00%	24	24	0	0	/home/oscar.carballal/devel/ec
News	100,00%	23	23	0	0	/home/oscar.carballal/devel/ec
Proposals	100,00%	20	20	0	0	/home/oscar.carballal/devel/ec
Userprofile	38,00%	375	145	0	0	/home/oscar.carballal/devel/ec

Inglés						
Application	Progress	Mensajes	Translated	Fuzzy	Obsolete	File
Spaces	0,00%	27	0	0	0	/home/oscar.carballal/devel/ec
News	0,00%	9	0	0	0	/home/oscar.carballal/devel/ec
Userprofile	0,00%	375	0	0	0	/home/oscar.carballal/devel/ec

Haz clic en el componente que quieras traducir y empieza (la traducción se realiza desde el inglés a los demás idiomas). Si te atascas en algún momento puedes utilizar la opción “Sugerencia”, que hará una petición a Google Translate y pondrá la traducción más oportuna.

Warning: Nunca confíes en el resultado del botón “Sugerir”. En muchas ocasiones se equivocará.

4.5.2 Traduciendo con gettext

Gettext es una herramienta bien conocida por todo los traductores alrededor del mundo ya que es un estándar. Gracias al *middleware* de django, el trabajo será mínimo, sólo tendremos que editar los archivos .po que hay en el código fuente.

```
ooscar.carballal@oscar.carballal:~/devel/ecidadania/src/e_cidadania/apps/spaces/locale/es$ cd LC_MESSAGES/
ooscar.carballal@oscar.carballal:~/devel/ecidadania/src/e_cidadania/apps/spaces/locale/es/LC_MESSAGES$ ls
total 24K
-rwxrwxr-x 2 oscar.carballal oscar.carballal 4,0K 2011-03-18 12:44 .
-rwxrwxr-x 3 oscar.carballal oscar.carballal 4,0K 2011-03-18 12:44 ..
-rw-rw-r-- 1 oscar.carballal oscar.carballal 4,5K 2011-03-18 12:44 django.mo
-rw-rw-r-- 1 oscar.carballal oscar.carballal 7,0K 2011-03-18 12:44 django.po
ooscar.carballal@oscar.carballal:~/devel/ecidadania/src/e_cidadania/apps/spaces/locale/es/LC_MESSAGES$
```

Los ficheros suelen encontrarse en un directorio llamado **locale** dentro del módulo. En él podrás encontrar directorios con los códigos de país (es_ES, en_GB, gl_ES, etc.) y dentro de ellos los ficheros .PO y .MO.

Para traducir, debes editar el fichero .PO que es un fichero de texto plano. Los ficheros .MO son la versión compilada del .PO para que el ordenador pueda entenderlos.

Warning: Establecer un flujo de trabajo para los traductores y explicarlo.

Referencia (inglés)

5.1 `apps.ecidadania.cal` — Calendar and events

5.1.1 `cal.models` — Data models

The calendar module calls a version of Python HTML Calendar and adds some functions to use django objects with it.

The source code is based on the work of Eivind Uggedal <eivind@uggedal.com>

class `apps.ecidadania.cal.models.EventCalendar` (*LocaleHTMLCalendar*)

Event calendar is a basic calendar made with HTMLCalendar module and its instance LocaleHTMLCalendar for translation.

Attributes `LocaleHTMLCalendar`

Methods `formatday`, `formatmonth`, `group_by_day`, `day_cell`

day_cell (*cssclass*, *body*)

Create the day cell.

formatday (*day*, *weekday*)

Format the day cell with the current events for the day.

formatmonth (*year*, *month*)

Format the current month with the events.

group_by_day (*events*)

Group the returned events into their respective dates.

5.1.2 `cal.views` — Views

5.2 `apps.ecidadania.debate` — Debates

5.2.1 `debate.admin` — Administration

5.2.2 `debate.models` — Data models

5.2.3 `debate.forms` — Forms

5.2.4 `debate.views` — Views

5.3 `core.spaces` — Spaces/Processes

5.3.1 `spaces.admin` — Administration

5.3.2 `spaces.models` — Data models

5.3.3 `spaces.forms` — Forms

`EntityFormSet` — `modelformset_factory`

5.3.4 spaces.views — Views

Intent views

RSS views

News views

Spaces views

Document views

Event views

5.4 apps.ecidadania.proposals — Proposals

5.4.1 proposals.admin — Administration

5.4.2 proposals.models — Data models

5.4.3 proposals.forms — Forms

5.4.4 proposals.views — Views

5.5 apps.ecidadania.news — News

5.5.1 news.admin — Administration

5.5.2 news.models — Data models

5.5.3 news.forms — Forms

5.5.4 news.views — Views

Colofón

6.1 Conseguir ayuda

- Prueba a mirar en el [P+F](#) (Preguntas Más Frecuentes).
- Listas de correo de e-cidadania
 - ecidadania-users@freelists.org
 - ecidadania-dev@freelists.org
 - ecidadania-es@freelists.org
- Crear un informe de error en el [bug tracker](#) o infórmanos a través de correo electrónico en soporte@ecidadania.org

Python Module Index

a

`apps.ecidadania.cal.models`, [23](#)