
visoreducativochaco

Versión 1.0

Edgardo Javier Gómez

15 de octubre de 2024

Contents

1	Documentación de la app Archivar	1
1.1	Modelos	1
1.2	Vistas	6
1.3	Formularios	9
1.4	Mixins	9
2	Documentación de la app Renpe	11
2.1	Modelos	11
2.2	Vistas	34
2.3	Formularios	37
3	Documentación de la app Dashboard	43
3.1	Vistas	43
4	Documentación de la app Directores	45
4.1	Vistas	45
5	Documentación de la app Establecimientos	47
5.1	Modelos	47
5.2	Vistas	53
6	Documentación de la app Lectocomprensión	55
6.1	Vistas	55
7	Documentación de la app Login	59
7.1	Vistas	59
8	Documentación de la app Mapas	61
8.1	Modelos	61
8.2	Vista Mapa	63
8.3	Vista Colectivos	64
8.4	Vista Comisarias	64
8.5	Vista Centros Salud	65
8.6	Vista Cueradio	66
8.7	Formularios	67
9	Documentación de la app Mapoteca	69
9.1	Modelos	69

9.2	Vistas	70
10	Documentación de la app Normativa	71
10.1	Modelos	71
10.2	Vistas	72
11	Documentación de la app Mapas	75
11.1	Modelos	75
11.2	Vistas	85
11.3	Formularios	92
12	Documentación de la app Reportes	101
12.1	Modelos	101
12.2	Vistas	101
13	Documentación de la app Supervisores	107
13.1	Modelos	107
13.2	Vistas	110
13.3	Formularios	114
14	Documentación de la app Usuarios	119
14.1	Modelos	119
14.2	Vistas	123
14.3	Formularios	128
14.4	Backends	131
14.5	Mixins	132
15	Documentación de la app Videoteca	133
15.1	Vistas	133

1.1 Modelos

class apps.archivar.models.**ArchRegister**(*args, **kwargs)

Bases: Model

Modelo para registrar archivos normativos asociados a un anexo.

cueanexo

Código único del anexo, con un máximo de 10 caracteres.

Type

str

asunto

Relación con el modelo AsuntoRegister, que describe el asunto del registro.

Type

ForeignKey

nivel

Relación con el modelo Nivel, que indica el nivel de clasificación del registro.

Type

ForeignKey

t_norma

Relación con el modelo TNormativa, que describe el tipo de normativa.

Type

ForeignKey

nro_normativa

Número de la normativa correspondiente, con un máximo de 100 caracteres.

Type
str

anio

Año asociado con la normativa, debe ser un entero.

Type
int

descripcion

Descripción del archivo o su contenido.

Type
str

archivo

Archivo a ser subido y asociado al registro, con un directorio de carga especificado.

Type
FileField

ruta

Ruta donde se almacena el archivo subido, puede ser nula o vacía.

Type
str

nombre_asunto(*str*)

Devuelve el asunto del registro.

save()

Sobrescribe el método save para actualizar la ruta del archivo una vez guardado.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

archivo

The descriptor for the file attribute on the model instance. Return a FieldFile when accessed so you can write code like:

```
>>> from myapp.models import MyModel
>>> instance = MyModel.objects.get(pk=1)
>>> instance.file.size
```

Assign a file object on assignment so you can do:

```
>>> with open('/path/to/hello.world') as f:
...     instance.file = File(f)
```

asunto

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

asunto_id

año

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cueanexo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

descripcion

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nivel

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

nivel_id

property nombre_asunto

Obtiene el nombre del asunto relacionado con el registro.

Devuelve

Nombre del asunto, o una cadena vacía si no hay asunto.

Tipo del valor devuelto

str

nro_normativa

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

ruta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(*args, **kwargs)

Guarda el registro y actualiza la ruta del archivo subido.

Llama al método save de la superclase y luego actualiza el campo “ruta” con la ubicación del archivo una vez que ha sido guardado en el sistema.

Parámetros

- ***args** – Argumentos adicionales para el método save.
- ****kwargs** – Argumentos adicionales para el método save.

t_norma

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

t_norma_id

class apps.archivar.models.AsuntoRegister(*args, **kwargs)

Bases: Model

Modelo para almacenar asuntos relacionados con registros de archivos.

asunto

Descripción del asunto, con un máximo de 150 caracteres.

Type

str

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

archregister_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

asunto

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

class apps.archivar.models.TNormativa(*args, **kwargs)

Bases: Model

Modelo para almacenar tipos de normativa.

t_norma

Descripción del tipo de normativa, con un máximo de 150 caracteres.

Type

str

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

archregister_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

t_norma

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class apps.archivar.models.nivel(*args, **kwargs)

Bases: Model

Modelo para almacenar niveles de clasificación.

niveles

Nombre del nivel, con un máximo de 150 caracteres.

Type

str

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

archregister_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

niveles

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

1.2 Vistas

class apps.archivar.views.**ArchivoCreateView**(**kwargs)

Vista para crear un nuevo archivo normativo.

Hereda de GroupRequiredMixin para asegurar que solo los usuarios con el grupo adecuado pueden acceder a esta vista.

model

Modelo de datos para el registro de archivos.

Type

ArchRegister

form_class

Formulario utilizado para crear un archivo.

Type

ArchRegisterForm

template_name

Nombre del template a renderizar.

Type

str

success_url

URL a la que se redirige al usuario después de una creación exitosa.

Type

str

get_context_data()

Añade contexto adicional al template, como el título de la página.

form_class

alias de ArchRegisterForm

get_context_data(kwargs)**

Insert the form into the context dict.

model

alias de ArchRegister

success_url = '/cargar/listar/'

template_name = 'archivos/cargar_archivo.html'

class apps.archivar.views.ArchivosListView(kwargs)**

Vista para listar todos los archivos normativos.

Hereda de ReadOnlyAccessMixin para permitir el acceso solo de lectura a los usuarios autorizados.

model

Modelo de datos para el registro de archivos.

Type

ArchRegister

template_name

Nombre del template a renderizar.

Type

str

context_object_name

Nombre del contexto que contiene la lista de archivos.

Type

str

get_context_data()

Añade contexto adicional al template, como el título de la página.

context_object_name = 'archivos'

get_context_data(kwargs)**

Get the context for this view.

model

alias de ArchRegister

template_name = 'archivos/archivos_lista.html'

class apps.archivar.views.BuscarPDFView(kwargs)**

Vista para buscar un archivo PDF basado en cueanexo y asunto.

post()

Maneja las solicitudes POST para buscar el archivo y devolver su URL.

post(request, *args, **kwargs)

class apps.archivar.views.**EliminarArchivosView**(**kwargs)

Vista para eliminar un archivo normativo.

Hereda de GroupRequiredMixin para asegurar que solo los usuarios con el grupo adecuado pueden acceder a esta vista.

get()

Maneja las solicitudes GET para eliminar el archivo y redirigir al usuario.

get(request)

class apps.archivar.views.**editar_archivos**(**kwargs)

Vista para editar un archivo normativo existente.

Hereda de GroupRequiredMixin para asegurar que solo los usuarios con el grupo adecuado pueden acceder a esta vista.

model

Modelo de datos para el registro de archivos.

Type

ArchRegister

form_class

Formulario utilizado para editar el archivo.

Type

ArchRegisterForm

template_name

Nombre del template a renderizar.

Type

str

success_url

URL a la que se redirige al usuario después de una edición exitosa.

Type

str

get_object()

Obtiene el objeto ArchRegister que se va a editar.

get_context_data()

Añade contexto adicional al template, como el título de la página.

form_class

alias de ArchRegisterForm

get_context_data(**kwargs)

Insert the form into the context dict.

get_object()

Return the object the view is displaying.

Require *self.queryset* and a *pk* or *slug* argument in the URLconf. Subclasses can override this to return any object.

```

model
    alias de ArchRegister

success_url = '/cargar/listar/'

template_name = 'archivos/editar.html'

```

1.3 Formularios

```

class apps.archivar.forms.ArchRegisterForm(data=None, files=None, auto_id='id_%s', prefix=None,
                                             initial=None, error_class=<class
                                             'django.forms.utils.ErrorList'>, label_suffix=None,
                                             empty_permitted=False, instance=None,
                                             use_required_attribute=None, renderer=None)

```

```

class Meta

    fields = ['cueanexo', 'asunto', 'nivel', 't_norma', 'nro_normativa', 'año',
              'descripcion', 'archivo']

    model
        alias de ArchRegister

    widgets = {'descripcion': <django.forms.widgets.Textarea object>}

base_fields = {'archivo': <django.forms.fields.FileField object>, 'asunto':
<django.forms.models.ModelChoiceField object>, 'año':
<django.forms.fields.IntegerField object>, 'cueanexo':
<django.forms.fields.CharField object>, 'descripcion':
<django.forms.fields.CharField object>, 'nivel':
<django.forms.models.ModelChoiceField object>, 'nro_normativa':
<django.forms.fields.CharField object>, 't_norma':
<django.forms.models.ModelChoiceField object>}

declared_fields = {}

property media
    Return all media required to render the widgets on this form.

```

1.4 Mixins

```

class apps.archivar.mixins.GroupRequiredMixin

```

Mixin que restringe el acceso a las vistas solo a usuarios pertenecientes a grupos específicos.

Este mixin verifica si el usuario autenticado pertenece a los grupos “Administrador” o “Gestor”. Si el usuario no pertenece a estos grupos, se deniega el acceso y se lanza una excepción de Permiso Denegado.

```

test_func()
    Verifica si el usuario pertenece a alguno de los grupos permitidos.

handle_no_permission()
    Lanza una excepción de Permiso Denegado si el acceso es denegado.

```

handle_no_permission()

test_func()

class apps.archivar.mixins.ReadOnlyAccessMixin

handle_no_permission()

test_func()

Mixin que proporciona acceso solo de lectura a las vistas para usuarios pertenecientes a grupos específicos.

Este mixin permite que los usuarios autenticados que pertenezcan a los grupos “Administrador”, “Gestor”, “Director”, “Supervisor”, o “Regional” tengan acceso. Si el usuario no pertenece a ninguno de estos grupos, se deniega el acceso y se lanza una excepción de Permiso Denegado.

test_func()

Verifica si el usuario pertenece a alguno de los grupos permitidos.

handle_no_permission()

Lanza una excepción de Permiso Denegado si el acceso es denegado.

2.1 Modelos

```
class apps.cenpe.models.Academica_Cenpe(id, usuario, titulo, tipo_form, nivel_form, tipo_inst, gestion_inst,
                                         reg_nro, f_egreso)
```

Bases: `Model`

exception `DoesNotExist`

Bases: `ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `MultipleObjectsReturned`

f_egreso

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

gestion_inst

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

gestion_inst_id

```
get_next_by_f_egreso(*, field=<django.db.models.fields.DateField: f_egreso>, is_next=True, **kwargs)
```

```
get_previous_by_f_egreso(*, field=<django.db.models.fields.DateField: f_egreso>, is_next=False,
                           **kwargs)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nivel_form

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

nivel_form_id

objects = <django.db.models.manager.Manager object>

reg_nro

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(*args, **kwargs)

Save the current instance. Override this in a subclass if you want to control the saving process.

The “force_insert” and “force_update” parameters can be used to insist that the «save» must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

tipo_form

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

tipo_form_id**tipo_inst**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

tipo_inst_id**titulo**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

usuario

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class apps.cenpe.models.CargosHoras_Cenpe(id, usuario, cueanexo, categoria, jornada, zona,  
nivel_cargohora, cargos_horas, cant_horas, lunes, martes,  
miercoles, jueves, viernes, situacion_revista, funciones,  
condicion_actividad, fecha_desde, fecha_hasta, cuof,  
cuof_anexo)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

cant_horas

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cargos_horas

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

categoria

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

condicion_actividad

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cueanexo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cuof

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cuof_anexo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

fecha_desde

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

fecha_hasta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

funciones

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_fecha_desde(**field*=<*django.db.models.fields.DateField: fecha_desde*>, *is_next*=True, ***kwargs*)

get_next_by_fecha_hasta(**field*=<*django.db.models.fields.DateField: fecha_hasta*>, *is_next*=True, ***kwargs*)

get_previous_by_fecha_desde(**field*=<*django.db.models.fields.DateField: fecha_desde*>, *is_next*=False, ***kwargs*)

get_previous_by_fecha_hasta(**field*=<*django.db.models.fields.DateField: fecha_hasta*>, *is_next*=False, ***kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

jornada

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

jueves

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

lunes

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

martes

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

miercoles

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nivel_cargohora

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <*django.db.models.manager.Manager* object>

save(**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The “force_insert” and “force_update” parameters can be used to insist that the «save» must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

situacion_revista

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

usuario

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

viernes

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

zona

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class apps.cenpe.models.Categoria_Cueanexo(id, nom_categoria)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom_categoria

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.CeicPuntos(*args, **kwargs)
```

Bases: Model

Modelo que representa los puntos CEIC asignados a un nivel educativo.

Atributos:

nivel (str): Nivel educativo al que pertenece. ceic_id (int): ID único para CEIC. descripcion_ceic (str): Descripción detallada del CEIC. estado (bool): Estado activo o inactivo del registro. puntos (int): Puntos asignados para el nivel y CEIC específicos.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

ceic_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

descripcion_ceic

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

estado

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nivel

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

puntos

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class apps.cenpe.models.Datos_Personal_Cenpe(id, usuario, apellidos, nombres, pais_nac, nacionalidad,  
t_doc, dni, cuil, sexo, estado_civil, nivel_form, telfijo,  
celular, prov_nac, loc_nac, f_nac, prov_resid, loc_resid,  
calle, nro, mz, pc, casa, piso, uf, barrio,  
id_jurisdiccional)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

apellidos

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

barrio

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

calle

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

casa

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

celular

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

clean()

Hook for doing any extra model-wide validation after clean() has been called on every field by self.clean_fields. Any ValidationError raised by this method will not be associated with a particular field; it will have a special-case association with the field defined by NON_FIELD_ERRORS.

cuil

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

dni

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

estado_civil

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

estado_civil_id

f_nac

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_f_nac(* , field=<django.db.models.fields.DateField: f_nac>, is_next=True, **kwargs)

get_previous_by_f_nac(* , field=<django.db.models.fields.DateField: f_nac>, is_next=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id_jurisdiccional

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

loc_nac

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

loc_resid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

mz

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nacionalidad

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

nacionalidad_id

nivel_form

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

nivel_form_id

nombres

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nro

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

pais_nac

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

pais_nac_id

pc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

piso

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

prov_nac

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

prov_resid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(*args, **kwargs)

Save the current instance. Override this in a subclass if you want to control the saving process.

The “force_insert” and “force_update” parameters can be used to insist that the «save» must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

sexo

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

sexo_id**t_doc**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

t_doc_id**telfijo**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

uf

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

usuario

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class apps.cenpe.models.Estado_Civil_Cenpe(id, descripcion_estciv)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

datos_personal_cenpe_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

descripcion_estciv

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.Estado_Titulo_Cenpe(id, estado_titulo)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

estado_titulo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.Gestion_Institucion_Cenpe(id, t_gestion)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

academica_cenpe_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

t_gestion

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class apps.cenpe.models.Nivel_Formacion_Cenpe(id, nivel_form)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

academica_cenpe_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

datos_personal_cenpe_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nivel_form

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.Nivel_Sistema(id, niv_sis)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

niv_sis

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.PadronCenpe(id, cueanexo, id_establecimiento, id_localizacion, id_oferta_local,
                                     nom_est, acronimo_oferta, oferta, nro_est, ambito, sector,
                                     region_loc, ref_loc, calle, numero, localidad, departamento,
                                     estado_loc, est_oferta, estado_est, jornada)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

acronimo_oferta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ambito

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

calle

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cueanexo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

departamento

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

est_oferta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

estado_est

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

estado_loc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id_establecimiento

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id_localizacion

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id_oferta_local

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

jornada

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

localidad

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom_est

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nro_est

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

numero

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

oferta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ref_loc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

region_loc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sector

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class apps.cenpe.models.SituacionRevista(id, sit_rev)

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

sit_rev

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class apps.cenpe.models.TipoJornada_Cueanexo(id, tipo_jornada)
    Bases: Model

    exception DoesNotExist
        Bases: ObjectDoesNotExist

    exception MultipleObjectsReturned
        Bases: MultipleObjectsReturned

    id
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
        executed.

    objects = <django.db.models.manager.Manager object>

    tipo_jornada
        A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
        executed.
```

```
class apps.cenpe.models.TipoFormacion_Cenpe(id, t_form)
    Bases: Model

    exception DoesNotExist
        Bases: ObjectDoesNotExist

    exception MultipleObjectsReturned
        Bases: MultipleObjectsReturned

    academica_cenpe_set
        Accessor to the related objects manager on the reverse side of a many-to-one relation.

        In the example:
```

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
id
    A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
    executed.

objects = <django.db.models.manager.Manager object>

t_form
    A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is
    executed.
```

```
class apps.cenpe.models.TipoInstitucion_Cenpe(id, t_inst)
    Bases: Model

    exception DoesNotExist
        Bases: ObjectDoesNotExist

    exception MultipleObjectsReturned
        Bases: MultipleObjectsReturned
```

academica_cenpe_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

t_inst

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class apps.cenpe.models.Tipo_Trayectoria(id, t_trayecto)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ingreso_trayectoria

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

objects = <django.db.models.manager.Manager object>

t_trayecto

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

trayectoria_ocupacional_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

```
class apps.cenpe.models.Trayectoria_Ocupacional(id, usuario, ingreso_tray, actual_tray, f_ingreso,
                                                frente_alumno, anios_frentealumn, apoyo,
                                                anios_apoyo, func_dir, anios_dir, act_extraesc,
                                                cant_hs)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

act_extraesc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

actual_tray

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

actual_tray_id

anios_apoyo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

anios_dir

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

anios_frentealumn

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

apoyo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cant_hs

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

f_ingreso

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

frente_alumno

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

func_dir

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_f_ingreso(**, field=<django.db.models.fields.DateField: f_ingreso>, is_next=True, **kwargs*)

get_previous_by_f_ingreso(**, field=<django.db.models.fields.DateField: f_ingreso>, is_next=False, **kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ingreso_tray

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

ingreso_tray_id

objects = <django.db.models.manager.Manager object>

usuario

Accessor to the related object on the forward side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Restaurant.place is a ForwardOneToOneDescriptor instance.

usuario_id

class apps.cenpe.models.Zona_Cueanexo(*id, tipo_zona*)

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

tipo_zona

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class apps.cenpe.models.condicionactividad(*id, cond_act*)

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

cond_act

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

class apps.cenpe.models.documento_tipo(*args, **kwargs)

Bases: Model

Modelo para representar los diferentes tipos de documentos de identidad.

Atributos:

c_tipo (int): Código único para el tipo de documento. descripcion_doc (str): Descripción del tipo de documento.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

c_tipo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

datos_personal_cenpe_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

descripcion_doc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.funciones(id,function)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

function

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.grado_tipo(*args, **kwargs)
```

Bases: Model

Modelo para representar los tipos de grados académicos.

Atributos:

c_grado (int): Código único del grado académico. descripcion_grado (str): Descripción del grado académico.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

c_grado

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

descripcion_grado

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.localidad_tipo(*args, **kwargs)
```

Bases: Model

Modelo que representa las diferentes localidades.

Atributos:

c_localidad (int): Código único de la localidad. descripcion_loc (str): Descripción de la localidad.
c_departamento (int): Código del departamento asociado. descripcion_dpto (str): Descripción del departamento. c_provincia (ForeignKey): Relación con el modelo *provincia_tipo*.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

c_departamento

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

c_localidad

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

c_provincia

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

c_provincia_id**descripcion_dpto**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

descripcion_loc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

class `apps.cenpe.models.nacionalidad(*args, **kwargs)`

Bases: `Model`

Modelo para representar las diferentes nacionalidades.

Atributos:

`c_nacionalidad` (int): Código único de la nacionalidad. `descripcion_nacional` (str): Descripción de la nacionalidad. `c_pais` (`ForeignKey`): Relación con el modelo *pais*.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

c_nacionalidad

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

c_pais

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

c_pais_id**datos_personal_cenpe_set**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

descripcion_nacional

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.oferta_tipo(c_oferta, descripcion_ofer)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

c_oferta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

descripcion_ofer

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.orientacion_tipo(c_orientacion, descripcion_orien)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

c_orientacion

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

descripcion_orien

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

class apps.cenpe.models.pais(*args, **kwargs)

Bases: Model

Modelo para representar los diferentes países.

Atributos:

c_pais (int): Código único del país. descripcion_pais (str): Nombre del país.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

c_pais

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

datos_personal_cenpe_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

descripcion_pais

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nacionalidad_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.provincia_tipo(*args, **kwargs)
```

Bases: Model

Modelo que representa las diferentes provincias.

Atributos:

c_provincia (int): Código único para la provincia. descripcion_prov (str): Descripción de la provincia.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

c_provincia

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

descripcion_prov

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

localidad_tipo_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

objects = <django.db.models.manager.Manager object>

```
class apps.cenpe.models.sexo_tipo(c_sexo, descripcion_sex)
```

Bases: Model

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

c_sexo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

datos_personal_cenpe_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

descripcion_sex

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

2.2 Vistas

class `apps.cenpe.views.CargosHorasCenpeCreateView(**kwargs)`

Vista basada en clases para gestionar la creación de los datos laborales del docente para el RENPEE.

Atributos:

`model` (Model): El modelo relacionado con los datos laborales (`CargosHoras_Cenpe`). `form_class` (Form): El formulario que se utiliza para gestionar la creación de los datos laborales. `template_name` (str): El template que se renderiza para la creación de datos laborales. `success_url` (str): La URL de redirección después de la creación exitosa de los datos laborales.

Métodos:

`get_initial`: Inicializa el campo “usuario” con el nombre de usuario del usuario autenticado. `form_valid`: Asigna el nombre de usuario al campo “usuario” antes de guardar los datos.

form_class

alias de `CargosHorasCenpeForm`

form_valid(form)

If the form is valid, save the associated model.

get_initial()

Return the initial data to use for forms on this view.

model

alias de `CargosHoras_Cenpe`

success_url = `'/censo/listado/'`

template_name = `'cenpe/cargar_cargoshoras.html'`

class `apps.cenpe.views.CargosHorasCenpeListView(**kwargs)`

Vista basada en clases para mostrar una lista de los cargos y horas del docente en el RENPEE.

Atributos:

`model` (Model): El modelo relacionado con los cargos y horas (`CargosHoras_Cenpe`). `template_name` (str): El template que se renderiza para mostrar la lista de cargos y horas. `context_object_name` (str): El nombre de la variable de contexto para acceder a los datos en la plantilla.

Métodos:

`get_queryset`: Filtra la lista de cargos y horas para mostrar solo los relacionados con el usuario autenticado. `get_context_data`: Añade el título al contexto de la plantilla.

context_object_name = `'Cargos_Horas'`

get_context_data(kwargs)**

Get the context for this view.

get_queryset()

Return the list of items for this view.

The return value must be an iterable and may be an instance of *QuerySet* in which case *QuerySet* specific behavior will be enabled.

model

alias de CargosHoras_Cenpe

template_name = 'cenpe/listadocargoshorascenpe.html'

class apps.cenpe.views.DatosAcademicosCenpeCreateView(kwargs)**

Vista basada en clases para gestionar la creación de datos académicos del docente para el RENPEE.

Atributos:

model (Model): El modelo relacionado con los datos académicos (Academica_Cenpe). form_class (Form): El formulario que se utiliza para gestionar la creación de los datos académicos. template_name (str): El template que se renderiza para la creación de datos académicos. success_url (str): La URL de redirección después de la creación exitosa de los datos académicos.

Métodos:

get_initial: Inicializa el campo “usuario” con el nombre de usuario del usuario autenticado. form_valid: Asigna el nombre de usuario al campo “usuario” antes de guardar los datos.

form_class

alias de DatosAcademicosCenpeForm

form_valid(form)

If the form is valid, save the associated model.

get_initial()

Return the initial data to use for forms on this view.

model

alias de Academica_Cenpe

success_url = '/censo/listado/'

template_name = 'cenpe/crear_datos_academicos.html'

class apps.cenpe.views.DatosPersonalCenpeCreateView(kwargs)**

Vista basada en clases para gestionar la creación de datos personales de los docentes para el RENPEE.

Atributos:

model (Model): El modelo relacionado con los datos personales (Datos_Personal_Cenpe). form_class (Form): El formulario que se utiliza para gestionar la creación de los datos personales. template_name (str): El template que se renderiza para la creación de datos personales. success_url (str): La URL de redirección después de la creación exitosa de los datos.

Métodos:

get_initial: Inicializa el campo “usuario” con el nombre de usuario del usuario autenticado. form_valid: Asigna el nombre de usuario al campo “usuario” antes de guardar los datos.

form_class

alias de DatosPersonalCenpeForm

form_valid(*form*)

If the form is valid, save the associated model.

get_initial()

Return the initial data to use for forms on this view.

model

alias de Datos_Personal_Cenpe

success_url = '/censo/academico/'

template_name = 'cenpe/crear_datos_personales.html'

class apps.cenpe.views.**EliminarDocentesView**(***kwargs*)

Vista basada en funciones para eliminar los cargos y horas del docente en el RENPEE.

Métodos:

get: Captura el ID del docente y lo elimina de la base de datos si existe.

get(*request*)

apps.cenpe.views.**IndexCenpe**(*request*)

Renderiza la página principal del módulo RENPEE para el usuario.

Parámetros

request (*HttpRequest*) – La solicitud HTTP que contiene la información del usuario.

Devuelve

Renderiza el template “cenpe/indexcenpe.html”.

Tipo del valor devuelto

HttpResponse

apps.cenpe.views.**cargar_localidades**(*request*)

Carga las localidades en función de la provincia seleccionada a través de una solicitud AJAX.

Parámetros

request (*HttpRequest*) – La solicitud HTTP que contiene el parámetro “provincia_id”.

Devuelve

Retorna un JSON con las localidades filtradas por la provincia seleccionada.

Tipo del valor devuelto

JsonResponse

apps.cenpe.views.**obtener_cargos_por_nivel**(*request*)

Filtra los cargos disponibles según el nivel educativo seleccionado.

Parámetros

request (*HttpRequest*) – La solicitud HTTP que contiene el parámetro “nivel”.

Devuelve

Un JSON con los cargos disponibles filtrados por el nivel seleccionado.

Tipo del valor devuelto

JsonResponse

apps.cenpe.views_reportes.**GenerarCertificado**(*request*)

Genera un certificado en formato PDF para un usuario autenticado, mostrando sus datos personales y cargos docentes.

Este certificado incluye: - Encabezado con información institucional. - Detalles de los datos personales del usuario, obtenidos de la base de datos. - Una tabla con los cargos y horas trabajadas. - Un código QR que contiene los datos completos del certificado.

Parámetros

request (*HttpRequest*) – Objeto de la solicitud HTTP que contiene información del usuario autenticado.

Devuelve

Un archivo PDF con el certificado generado.

Tipo del valor devuelto

HttpResponse

Funcionalidad: 1. Obtiene los datos personales y los cargos del usuario autenticado a partir de una consulta SQL a la base de datos. 2. Genera un PDF con el módulo *reportlab*, incluyendo los datos personales y una tabla de cargos. 3. Genera un código QR que contiene la información del certificado. 4. Envía el archivo PDF generado como respuesta HTTP.

Ejemplo:

- El usuario autenticado accede a la vista, y se genera un PDF con su certificado.
- El código QR incrustado contiene información clave del certificado, como nombre, DNI, título, cargos y fechas.

Uso:

Esta vista se puede integrar como parte de un sistema que genere certificados para docentes o personal administrativo.

Notas:

- La consulta SQL es específica a la base de datos *cenpe.certificado_cenpe*, asegurando que los datos se obtengan según el *dni* del usuario.
- Se utiliza el módulo *qrcode* para generar un código QR y *reportlab* para generar el PDF.

2.3 Formularios

```
class apps.cenpe.forms.CargosHorasCenpeForm(*args, **kwargs)
```

Formulario para capturar los datos de cargos y horas en el sistema CENPE.

Campos:

cueanexo (*ModelChoiceField*): Campo de selección para el Cueanexo. *categoria* (*ModelChoiceField*): Campo de selección para la categoría de la escuela. *jornada* (*ModelChoiceField*): Campo de selección para el tipo de jornada. *zona* (*ModelChoiceField*): Campo de selección para la zona. *nivel_cargohora* (*ModelChoiceField*): Campo de selección para el nivel del cargo o la hora cátedra. *cargos_horas* (*ModelChoiceField*): Campo de selección para los cargos u horas cátedra. *cant_horas* (*DecimalField*): Campo numérico para ingresar la cantidad de horas. *lunes, martes, miercoles, jueves, viernes* (*BooleanField*): Checkboxes para indicar los días de trabajo. *situacion_revista* (*ModelChoiceField*): Campo de selección para la situación de revista. *funciones* (*ModelChoiceField*): Campo de selección para las funciones. *condicion_actividad* (*ModelChoiceField*): Campo de selección para la condición de actividad. *fecha_desde, fecha_hasta* (*DateField*): Campos de fecha para indicar el período. *cuof, cuof_anexo* (*IntegerField*): Campos numéricos para ingresar el CUOF y CUOF anexo.

Métodos:

__init__(): Inicializa el formulario, estableciendo el foco y personalizando las etiquetas.

label_from_instance_cueanexo(): Método personalizado para mostrar el Cueanexo y el nombre de la institución.

class Meta

fields = '__all__'

model

alias de CargosHoras_Cenpe

widgets = {'usuario': <django.forms.widgets.HiddenInput object>}

**base_fields = {'cant_horas': <django.forms.fields.DecimalField object>,
'cargos_horas': <django.forms.models.ModelChoiceField object>, 'categoria':
<django.forms.models.ModelChoiceField object>, 'condicion_actividad':
<django.forms.models.ModelChoiceField object>, 'cueanexo':
<django.forms.models.ModelChoiceField object>, 'cuof':
<django.forms.fields.IntegerField object>, 'cuof_anexo':
<django.forms.fields.IntegerField object>, 'fecha_desde':
<django.forms.fields.DateField object>, 'fecha_hasta':
<django.forms.fields.DateField object>, 'funciones':
<django.forms.models.ModelChoiceField object>, 'jornada':
<django.forms.models.ModelChoiceField object>, 'jueves':
<django.forms.fields.BooleanField object>, 'lunes':
<django.forms.fields.BooleanField object>, 'martes':
<django.forms.fields.BooleanField object>, 'miercoles':
<django.forms.fields.BooleanField object>, 'nivel_cargohora':
<django.forms.models.ModelChoiceField object>, 'situacion_revista':
<django.forms.models.ModelChoiceField object>, 'usuario':
<django.forms.fields.CharField object>, 'viernes':
<django.forms.fields.BooleanField object>, 'zona':
<django.forms.models.ModelChoiceField object>}**

**declared_fields = {'cant_horas': <django.forms.fields.DecimalField object>,
'cargos_horas': <django.forms.models.ModelChoiceField object>, 'categoria':
<django.forms.models.ModelChoiceField object>, 'condicion_actividad':
<django.forms.models.ModelChoiceField object>, 'cueanexo':
<django.forms.models.ModelChoiceField object>, 'cuof':
<django.forms.fields.IntegerField object>, 'cuof_anexo':
<django.forms.fields.IntegerField object>, 'fecha_desde':
<django.forms.fields.DateField object>, 'fecha_hasta':
<django.forms.fields.DateField object>, 'funciones':
<django.forms.models.ModelChoiceField object>, 'jornada':
<django.forms.models.ModelChoiceField object>, 'jueves':
<django.forms.fields.BooleanField object>, 'lunes':
<django.forms.fields.BooleanField object>, 'martes':
<django.forms.fields.BooleanField object>, 'miercoles':
<django.forms.fields.BooleanField object>, 'nivel_cargohora':
<django.forms.models.ModelChoiceField object>, 'situacion_revista':
<django.forms.models.ModelChoiceField object>, 'viernes':
<django.forms.fields.BooleanField object>, 'zona':
<django.forms.models.ModelChoiceField object>}**

label_from_instance_cueanexo(obj)

Devuelve la etiqueta personalizada para el campo *cueanexo* mostrando el Cueanexo y el nombre de la institución.

Parámetros

obj – Instancia de PadronCenpe.

Devuelve

Texto formateado que muestra el Cueanexo y el nombre de la institución.

Tipo del valor devuelto

str

property media

Return all media required to render the widgets on this form.

```
class apps.cenpe.forms.DatosAcademicosCenpeForm(*args, **kwargs)
```

Formulario para capturar los datos académicos del usuario en el sistema CENPE.

Campos:

titulo (CharField): Campo de texto para el título académico. tipo_form (ModelChoiceField): Campo de selección para el tipo de formación. nivel_form (ModelChoiceField): Campo de selección para el nivel de formación alcanzado. tipo_inst (ModelChoiceField): Campo de selección para el tipo de institución. gestion_inst (ModelChoiceField): Campo de selección para la gestión de la institución. reg_nro (CharField): Campo de texto para el número de registro del título. f_egreso (DateField): Campo de fecha para la fecha de egreso.

Métodos:

clean_titulo(): Limpia y valida el campo de título. __init__(): Inicializa el formulario y establece el foco en el campo título.

class Meta

```
fields = '__all__'
```

model

alias de Academica_Cenpe

```
widgets = {'usuario': <django.forms.widgets.HiddenInput object>}
```

```
base_fields = {'f_egreso': <django.forms.fields.DateField object>, 'gestion_inst':
<django.forms.models.ModelChoiceField object>, 'nivel_form':
<django.forms.models.ModelChoiceField object>, 'reg_nro':
<django.forms.fields.CharField object>, 'tipo_form':
<django.forms.models.ModelChoiceField object>, 'tipo_inst':
<django.forms.models.ModelChoiceField object>, 'titulo':
<django.forms.fields.CharField object>, 'usuario': <django.forms.fields.CharField
object>}
```

clean_titulo()

Asegura que el campo de título esté en mayúsculas y solo contenga caracteres válidos.

Devuelve

Título validado y convertido a mayúsculas.

Tipo del valor devuelto

str

```
declared_fields = {'f_egreso': <django.forms.fields.DateField object>,
'gestion_inst': <django.forms.models.ModelChoiceField object>, 'nivel_form':
<django.forms.models.ModelChoiceField object>, 'reg_nro':
<django.forms.fields.CharField object>, 'tipo_form':
<django.forms.models.ModelChoiceField object>, 'tipo_inst':
<django.forms.models.ModelChoiceField object>, 'titulo':
<django.forms.fields.CharField object>}
```

property media

Return all media required to render the widgets on this form.

```
class apps.cenpe.forms.DatosPersonalCenpeForm(*args, **kwargs)
```

Formulario para capturar los datos personales del usuario del sistema CENPE.

Campos:

apellidos (CharField): Campo de texto para el apellido. nombres (CharField): Campo de texto para el nombre en mayúsculas. dni (CharField): Campo de texto para ingresar el DNI. cuil (CharField): Campo de texto para ingresar el CUIL. telfijo (CharField): Campo de texto para ingresar el teléfono fijo. celular (CharField): Campo de texto para ingresar el número de celular. f_nac (DateField): Campo para ingresar la fecha de nacimiento. calle (CharField): Campo de texto para la calle de la dirección. nro (CharField): Campo de texto para el número de la dirección. mz (CharField): Campo de texto para el bloque o manzana. pc (CharField): Campo de texto para el código postal. casa (CharField): Campo de texto para el número de casa. piso (CharField): Campo de texto para el piso de la dirección. uf (CharField): Campo de texto para la unidad funcional. barrio (CharField): Campo de texto para el barrio. pais_nac (ModelChoiceField): Campo de selección para el país de nacimiento. nacionalidad (ModelChoiceField): Campo de selección para la nacionalidad. t_doc (ModelChoiceField): Campo de selección para el tipo de documento. sexo (ModelChoiceField): Campo de selección para el sexo. estado_civil (ModelChoiceField): Campo de selección para el estado civil. nivel_form (ModelChoiceField): Campo de selección para el nivel de formación alcanzado. prov_nac (ModelChoiceField): Campo de selección para la provincia de nacimiento. prov_resid (ModelChoiceField): Campo de selección para la provincia de residencia. loc_nac (ModelChoiceField): Campo de selección para la localidad de nacimiento. loc_resid (ModelChoiceField): Campo de selección para la localidad de residencia.

Métodos:

clean_apellidos(): Limpia y valida el campo de apellidos. clean_nombres(): Limpia y valida el campo de nombres. __init__(): Inicializa el formulario y carga dinámicamente las localidades según la provincia seleccionada.

class Meta

```
fields = '__all__'
```

model

```
alias de Datos_Personal_Cenpe
```

```
widgets = {'usuario': <django.forms.widgets.HiddenInput object>}
```

```

base_fields = {'apellidos': <django.forms.fields.CharField object>, 'barrio':
<django.forms.fields.CharField object>, 'calle': <django.forms.fields.CharField
object>, 'casa': <django.forms.fields.CharField object>, 'celular':
<django.forms.fields.CharField object>, 'cuil': <django.forms.fields.CharField
object>, 'dni': <django.forms.fields.CharField object>, 'estado_civil':
<django.forms.models.ModelChoiceField object>, 'f_nac':
<django.forms.fields.DateField object>, 'id_jurisdiccional':
<django.forms.fields.IntegerField object>, 'loc_nac':
<django.forms.models.ModelChoiceField object>, 'loc_resid':
<django.forms.models.ModelChoiceField object>, 'mz': <django.forms.fields.CharField
object>, 'nacionalidad': <django.forms.models.ModelChoiceField object>,
'nivel_form': <django.forms.models.ModelChoiceField object>, 'nombres':
<django.forms.fields.CharField object>, 'nro': <django.forms.fields.CharField
object>, 'pais_nac': <django.forms.models.ModelChoiceField object>, 'pc':
<django.forms.fields.CharField object>, 'piso': <django.forms.fields.CharField
object>, 'prov_nac': <django.forms.models.ModelChoiceField object>, 'prov_resid':
<django.forms.models.ModelChoiceField object>, 'sexo':
<django.forms.models.ModelChoiceField object>, 't_doc':
<django.forms.models.ModelChoiceField object>, 'telfijo':
<django.forms.fields.CharField object>, 'uf': <django.forms.fields.CharField
object>, 'usuario': <django.forms.fields.CharField object>}

```

clean_apellidos()

Asegura que el campo de apellidos esté en mayúsculas y solo contenga caracteres válidos.

Devuelve

Apellido validado y convertido a mayúsculas.

Tipo del valor devuelto

str

clean_nombres()

Asegura que el campo de nombres esté en mayúsculas y solo contenga caracteres válidos.

Devuelve

Nombre validado y convertido a mayúsculas.

Tipo del valor devuelto

str

```
declared_fields = {'apellidos': <django.forms.fields.CharField object>, 'barrio':  
<django.forms.fields.CharField object>, 'calle': <django.forms.fields.CharField  
object>, 'casa': <django.forms.fields.CharField object>, 'celular':  
<django.forms.fields.CharField object>, 'cuil': <django.forms.fields.CharField  
object>, 'dni': <django.forms.fields.CharField object>, 'estado_civil':  
<django.forms.models.ModelChoiceField object>, 'f_nac':  
<django.forms.fields.DateField object>, 'loc_nac':  
<django.forms.models.ModelChoiceField object>, 'loc_resid':  
<django.forms.models.ModelChoiceField object>, 'mz': <django.forms.fields.CharField  
object>, 'nacionalidad': <django.forms.models.ModelChoiceField object>,  
'nivel_form': <django.forms.models.ModelChoiceField object>, 'nombres':  
<django.forms.fields.CharField object>, 'nro': <django.forms.fields.CharField  
object>, 'pais_nac': <django.forms.models.ModelChoiceField object>, 'pc':  
<django.forms.fields.CharField object>, 'piso': <django.forms.fields.CharField  
object>, 'prov_nac': <django.forms.models.ModelChoiceField object>, 'prov_resid':  
<django.forms.models.ModelChoiceField object>, 'sexo':  
<django.forms.models.ModelChoiceField object>, 't_doc':  
<django.forms.models.ModelChoiceField object>, 'telfijo':  
<django.forms.fields.CharField object>, 'uf': <django.forms.fields.CharField  
object>}
```

property media

Return all media required to render the widgets on this form.

Documentación de la app Dashboard

3.1 Vistas

`apps.dashboard.views.dash(request)`

Renderiza la vista del dashboard para el usuario autenticado.

Si el usuario está autenticado, se obtiene su nivel de acceso. De lo contrario, se establece un valor por defecto (None).

Parámetros

request – El objeto `HttpRequest` que contiene la información sobre la solicitud realizada por el usuario.

Devuelve

Renderiza el template “dashboard/body.html” con el
contexto que incluye el nivel de acceso del usuario.

Tipo del valor devuelto

`HttpResponse`

`apps.dashboard.views.directores(request)`

Renderiza la vista institucional para directores.

Esta función renderiza el template “directores/institucional.html”, donde se presenta la información relevante para los directores.

Parámetros

request – El objeto `HttpRequest` que contiene la información sobre la solicitud realizada por el usuario.

Devuelve

Renderiza el template “directores/institucional.html”.

Tipo del valor devuelto

`HttpResponse`

`apps.dashboard.views.evaluacion(request)`

Renderiza la vista de evaluación.

Esta función renderiza el template “dashboard/portadaevaluacion.html”, donde se presenta la información relevante para la evaluación.

Parámetros

request – El objeto `HttpRequest` que contiene la información sobre la solicitud realizada por el usuario.

Devuelve

Renderiza el template “dashboard/portadaevaluacion.html”.

Tipo del valor devuelto

`HttpResponse`

`apps.dashboard.views.portada(request)`

Redirige a los usuarios autenticados a la vista correspondiente según su grupo, o renderiza la portada si no pertenecen a ningún grupo.

Si el usuario está autenticado y pertenece a un grupo específico (“Director”, “Evaluacion”, “Aplicador”), se redirige a la URL correspondiente. Si no pertenece a ninguno de esos grupos, se renderiza la portada.

Parámetros

request – El objeto `HttpRequest` que contiene la información sobre la solicitud realizada por el usuario.

Devuelve

Redirige a la URL correspondiente según el grupo del
usuario o renderiza el template “dashboard/portada.html”.

Tipo del valor devuelto

`HttpResponse`

Documentación de la app Directores

4.1 Vistas

`apps.directores.views.filter_matricula_views_directores(request)`

Filtra y muestra la matrícula de acuerdo a la oferta educativa.

Esta vista se encarga de obtener la matrícula de los estudiantes en función de la oferta educativa activa asociada al usuario logueado. Se conecta a la base de datos “visualizador” para realizar las consultas necesarias y luego renderiza la plantilla correspondiente con los resultados.

Parámetros

request – El objeto HttpRequest que contiene información sobre la solicitud del usuario.

Devuelve

Renderiza la plantilla “directores/matricula.html”

con los detalles de matrícula obtenidos de la base de datos.

Tipo del valor devuelto

HttpResponse

`apps.directores.views.filtrar_tablas_view_directores(request)`

Muestra los datos institucionales del usuario autenticado.

Esta vista se encarga de obtener y mostrar información relacionada con el usuario logueado, incluyendo datos institucionales, planes de estudio, anexos y ofertas. Se conecta a la base de datos “Padron” para realizar las consultas necesarias.

Parámetros

request – El objeto HttpRequest que contiene información sobre la solicitud del usuario.

Devuelve

Renderiza la plantilla “directores/institucional.html”

con los resultados obtenidos de la base de datos.

Tipo del valor devuelto
HttpResponse

Documentación de la app Establecimientos

5.1 Modelos

```
class apps.establecimientos.models.PadronActualizar(*args, **kwargs)
```

Bases: `Model`

Modelo que representa la información actualizada del padrón de establecimientos educativos.

Atributos:

`id_establecimiento` (`IntegerField`): Identificador único del establecimiento. `id_localizacion` (`IntegerField`): Identificador de la localización del establecimiento. `cueanexo` (`TextField`): Código único de establecimiento anexo. `nom_est` (`CharField`): Nombre del establecimiento. `nro_est` (`CharField`): Número del establecimiento. `anio_creac_establec` (`CharField`): Año de creación del establecimiento. `fecha_creac_establec` (`CharField`): Fecha de creación del establecimiento. `region` (`CharField`): Región geográfica del establecimiento. `udt` (`CharField`): Código UDT (Unidad de Decisión Territorial). `cui` (`CharField`): Código único de identificación. `cua` (`CharField`): Código CUA (Clave Única de Autorización). `cuof` (`CharField`): Código CUOF (Clave Única de Oferta Formativa). `sector` (`CharField`): Sector educativo (público o privado). `ambito` (`CharField`): Ámbito educativo. `ref_loc` (`CharField`): Referencia local del establecimiento. `calle` (`CharField`): Dirección del establecimiento. `numero` (`CharField`): Número de la dirección. `localidad` (`CharField`): Localidad donde se encuentra el establecimiento. `departamento` (`CharField`): Departamento geográfico del establecimiento. `cod_postal` (`CharField`): Código postal del establecimiento. `categoria` (`CharField`): Categoría del establecimiento. `estado_est` (`CharField`): Estado del establecimiento. `estado_loc` (`CharField`): Estado de la localización. `telefono_cod_area` (`CharField`): Código de área del teléfono. `telefono_nro` (`CharField`): Número de teléfono del establecimiento. `per_funcionamiento` (`CharField`): Período de funcionamiento del establecimiento. `email_loc` (`CharField`): Correo electrónico de la localización. `sitio_web` (`CharField`): Sitio web del establecimiento. `cooperadora` (`CharField`): Indicación de si tiene cooperadora. `sede` (`BooleanField`): Indicación de si es sede. `permanencia` (`CharField`): Tipo de permanencia del establecimiento. `sede_adm` (`BooleanField`): Indicación de si es sede administrativa. `resploc_apellido` (`CharField`): Apellido del responsable local. `resploc_nombre` (`CharField`): Nombre del responsable local. `resploc_telefono` (`CharField`): Teléfono del responsable local. `resploc_doc` (`IntegerField`): Documento del responsable local. `resploc_email` (`CharField`): Correo electrónico del responsable local. `resploc_nacimiento` (`DateField`): Fecha de nacimiento del responsable

local. resploc_cuitcuil (CharField): CUIT/CUIL del responsable local. arancel (SmallIntegerField): Arancel del establecimiento (si aplica).

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

ambito

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

anio_creac_establec

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

arancel

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

calle

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

categoria

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cod_postal

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cooperadora

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cua

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cueanexo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cui

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cuof

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

departamento

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

email_loc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

estado_est

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

estado_loc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

fecha_creac_establec

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id_establecimiento

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id_localizacion

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

localidad

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom_est

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nro_est

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

numero

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

per_funcionamiento

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

permanencia

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ref_loc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

region

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

resploc_apellido

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

resploc_cuitcuil

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

resploc_doc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

resploc_email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

resploc_nacimiento

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

resploc_nombre

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

resploc_telefono

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sector

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sede

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sede_adm

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sitio_web

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

telefono_cod_area

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

telefono_nro

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

udt

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class apps.establecimientos.models.**PadronOfertas**(*args, **kwargs)

Bases: Model

Modelo que representa las ofertas educativas asociadas a un establecimiento.

Atributos:

cueanexo (IntegerField): Código único del establecimiento anexo. id_establecimiento (CharField): Identificador del establecimiento. id_localizacion (CharField): Identificador de la localización del establecimiento. id_oferta_local (CharField): Identificador único de la oferta educativa. nom_est (CharField): Nombre del establecimiento. acronimo_oferta (CharField): Acrónimo de la oferta educativa. oferta (CharField): Descripción de la oferta educativa. nro_est (CharField): Número del establecimiento. ambito (CharField): Ámbito educativo. sector (CharField): Sector educativo (público o privado). region_loc (CharField): Región geográfica de la localización. ref_loc (CharField): Referencia local del establecimiento. calle (CharField): Dirección del establecimiento. numero (CharField): Número de la dirección. localidad (CharField): Localidad donde se encuentra el establecimiento. departamento (CharField): Departamento geográfico del establecimiento. estado_loc (CharField): Estado de la localización. est_oferta (CharField): Estado de la oferta educativa. estado_est (CharField): Estado del establecimiento. jornada (CharField): Tipo de jornada educativa.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

acronimo_oferta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ambito

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

calle

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cueanexo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

departamento

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

est_oferta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

estado_est

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

estado_loc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id_establecimiento

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id_localizacion

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id_oferta_local

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

jornada

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

localidad

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom_est

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nro_est

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

numero

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

oferta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ref_loc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

region_loc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sector

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

5.2 Vistas

`apps.establecimientos.views.establecimientos(request)`

Vista que filtra y muestra los establecimientos educativos activos.

Este método obtiene los establecimientos cuyo CUE anexo termina en “00” y cuya oferta educativa se encuentra activa. Los resultados son pasados al template *establecimientos/establecimientos.html* para su renderización.

Parámetros:

request (HttpRequest): La solicitud HTTP enviada por el usuario.

Retorna:

HttpResponse: Respuesta HTTP con la plantilla renderizada que incluye los establecimientos filtrados.

Documentación de la app Lectocomprensión

6.1 Vistas

`apps.lectocomp.views.calcular_estadisticas_por_cueanexo(username)`

Calcula estadísticas relacionadas con los estudiantes de una escuela específica (identificada por cueanexo).

Parámetros

username (*str*) – El cueanexo del usuario actual.

Devuelve

Incluye tres resultados:

- Lista de tuplas con el total de DNI por cueanexo y desempeño.
- Total de DNI sin discriminación de desempeño.
- Promedio de puntaje por cueanexo.

Tipo del valor devuelto

tuple

`apps.lectocomp.views.cargar_grafico_loc(request)`

Renderiza la plantilla “lectocomp/graficolocalidades.html” sin datos de gráficos inicialmente.

Parámetros

request (*HttpRequest*) – La solicitud HTTP recibida.

Devuelve

Respuesta renderizada con la plantilla “lectocomp/graficolocalidades.html”.

Tipo del valor devuelto

HttpResponse

`apps.lectocomp.views.cargar_grafico_reg(request)`

Renderiza la plantilla “lectocomp/graficoreg.html” sin datos de gráficos inicialmente.

Parámetros

request (*HttpRequest*) – La solicitud HTTP recibida.

Devuelve

Respuesta renderizada con la plantilla “lectocomp/graficoreg.html”.

Tipo del valor devuelto

HttpResponse

`apps.lectocomp.views.mostrar_grafico_loc(request)`

Muestra gráficos de desempeño académico basado en la localidad seleccionada. Filtra los datos por localidad y genera gráficos con los resultados.

Parámetros

request (*HttpRequest*) – La solicitud HTTP recibida.

Devuelve

Respuesta renderizada con la plantilla “lectocomp/graficolocalidades.html”,
que incluye gráficos y estadísticas filtradas por localidad.

Tipo del valor devuelto

HttpResponse

`apps.lectocomp.views.mostrar_grafico_localidad(request)`

Obtiene y muestra un listado de localidades con datos disponibles para la evaluación. Renderiza una vista que permite seleccionar localidades para filtrar datos de evaluación.

Parámetros

request (*HttpRequest*) – La solicitud HTTP recibida.

Devuelve

Respuesta renderizada con la plantilla “lectocomp/graficolocalidades.html”,
que incluye la lista de localidades.

Tipo del valor devuelto

HttpResponse

`apps.lectocomp.views.mostrar_grafico_reg(request)`

Filtra y muestra gráficos basados en las regiones seleccionadas por el usuario. Si no se selecciona ninguna región, muestra un mensaje indicando que no hay datos disponibles.

Parámetros

request (*HttpRequest*) – La solicitud HTTP recibida.

Devuelve

Respuesta renderizada con la plantilla “lectocomp/graficoreg.html”,
que incluye los gráficos y estadísticas calculadas.

Tipo del valor devuelto

HttpResponse

`apps.lectocomp.views.mostrar_pdf_informefinal(request)`

Renderiza una vista que muestra un PDF incrustado del informe final.

Parámetros

request (*HttpRequest*) – La solicitud HTTP recibida.

Devuelve

Respuesta renderizada con la plantilla “lectocomp/informefinal.html”.

Tipo del valor devuelto

HttpResponse

`apps.lectocomp.views.mostrar_pdf_recomendaciones(request)`

Renderiza una vista que muestra un PDF incrustado de recomendaciones.

Parámetros

request (*HttpRequest*) – La solicitud HTTP recibida.

Devuelve

Respuesta renderizada con la plantilla “lectocomp/recomendaciones.html”.

Tipo del valor devuelto

HttpResponse

`apps.lectocomp.views.tu_vista(request)`

Vista protegida que calcula y muestra las estadísticas por cueanexo del usuario actual. Presenta un gráfico de torta con las estadísticas calculadas y otros resultados relevantes.

Parámetros

request (*HttpRequest*) – La solicitud HTTP recibida.

Devuelve

Respuesta renderizada con la plantilla “lectocomp/grafico.html”.

Tipo del valor devuelto

HttpResponse

7.1 Vistas

class apps.login.views.CustomLogoutView(**kwargs)

Vista personalizada para gestionar el cierre de sesión.

Atributos:

next_page (str): URL a la que se redirige al usuario tras cerrar sesión.

next_page = '/portada/'

class apps.login.views.LoginFormView(**kwargs)

Vista personalizada para gestionar el inicio de sesión.

Atributos:

template_name (str): El nombre del template utilizado para el formulario de inicio de sesión.

Métodos:

get_context_data: Agrega el título “Iniciar Sesión” al contexto. get_success_url: Redirige al usuario autenticado según sus permisos y grupos, o a una URL predeterminada si no está en ningún grupo. form_valid: Valida el formulario de inicio de sesión y autentica al usuario. Retorna una respuesta JSON dependiendo del resultado. form_invalid: Retorna una respuesta JSON con un mensaje de error si las credenciales son incorrectas. post: Maneja la solicitud POST, incluyendo solicitudes AJAX, para validar el formulario y retornarlas respuestas correspondientes.

form_invalid(form)

Maneja el caso en que el formulario no es válido, retornando un mensaje de error.

Parámetros

form – Formulario de inicio de sesión inválido.

Devuelve

Respuesta JSON con el mensaje de error y el template asociado.

Tipo del valor devuelto

JsonResponse

form_valid(*form*)

Maneja el caso en que el formulario es válido, autenticando al usuario y redirigiéndolo según sus permisos.

Parámetros

form – Formulario de inicio de sesión.

Devuelve

Respuesta JSON con el estado de la autenticación.

Tipo del valor devuelto

JsonResponse

get_context_data(*kwargs*)**

Agrega el título “Iniciar Sesión” al contexto del template.

Parámetros

kwargs – Diccionario de argumentos adicionales.

Devuelve

Contexto actualizado con el título de la página.

Tipo del valor devuelto

dict

get_success_url()

Redirige al usuario autenticado basado en su grupo.

Devuelve

URL a la que será redirigido el usuario según su grupo.

Tipo del valor devuelto

str

post(*request*, **args*, *kwargs*)**

Procesa la solicitud POST, manejando solicitudes AJAX y no AJAX.

Parámetros

- **request** – Objeto de solicitud HTTP.
- ***args** – Argumentos adicionales.
- ****kwargs** – Argumentos adicionales.

Devuelve

Respuesta dependiendo de si la solicitud es AJAX o no.

Tipo del valor devuelto

JsonResponse o HttpResponseRedirect

template_name = 'login/login.html'

8.1 Modelos

class apps.mapas.models.LocalidadesRegion(*args, **kwargs)

Bases: Model

Modelo que representa los detalles de contacto y localización de las localidades dentro de una región.

Atributos:

reg (CharField): Código o nombre de la región. nom_dir (CharField): Nombre del director regional. tel_dir (CharField): Teléfono de contacto del director regional. email_dir (CharField): Correo electrónico del director regional. loc_reg (CharField): Nombre de la localidad correspondiente a la región. dep_reg (CharField): Nombre del departamento correspondiente a la región.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

dep_reg

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

email_dir

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

loc_reg

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom_dir

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

reg

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

tel_dir

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class apps.mapas.models.RegionalesGeometria(*args, **kwargs)

Bases: Model

Modelo que representa las geometrías de las regiones educativas.

Atributos:

id (AutoField): Identificador único del registro. geom (GeometryField): Campo espacial que almacena la geometría de la región. objectid (IntegerField): Identificador del objeto en la base de datos externa. region_pad (CharField): Nombre o código de la región. TITULO (TextField): Descripción o título de la región.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

TITULO

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

geom**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objectid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

region_pad

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

8.2 Vista Mapa

`apps.mapas.views.filter_data(request)`

`apps.mapas.views.filter_listado_map(request)`

`apps.mapas.views.filtrado(request)`

`apps.mapas.views.filtrado_list(request)`

`apps.mapas.views.filtrar_tablas_view(request)`

`apps.mapas.views.operaciones_comunes(request, template_name='publico/basecriterios.html')`

`apps.mapas.views2.mapapuntos(request)`

Renderiza la plantilla del mapa.

Esta vista simplemente carga la plantilla HTML “mapa/mapa.html”, donde se mostrará el mapa interactivo que puede luego cargar puntos a través de peticiones AJAX o a través de un contexto pasado desde otras vistas.

Parámetros

request – El objeto `HttpRequest` que contiene metadatos sobre la solicitud.

Devuelve

Renderiza la plantilla “mapa/mapa.html”.

Tipo del valor devuelto

`HttpResponse`

`apps.mapas.views2.obtener_datos_ofertas(request)`

Consulta y obtiene datos de ofertas educativas desde la base de datos.

Esta función realiza una consulta a la base de datos para obtener información sobre los establecimientos educativos y las ofertas asociadas, incluyendo nombre del establecimiento, tipo de oferta, ámbito, sector, y localidad. Los datos se envían al template “listadoestablecimientos.html”.

Parámetros

request – El objeto `HttpRequest` que contiene metadatos sobre la solicitud.

Devuelve

Renderiza la plantilla “mapa/listadoestablecimientos.html” con los datos de las ofertas. Si no se encuentran resultados, renderiza “consulta_vacia.html”.

Tipo del valor devuelto

`HttpResponse`

`apps.mapas.views2.obtenerdatos(request)`

Obtiene puntos geográficos de la base de datos y los convierte a formato GeoJSON.

Realiza una consulta a la base de datos para obtener los datos de las ofertas educativas, filtra las filas con coordenadas válidas, convierte los datos a formato GeoJSON y los envía a la plantilla “mapa/mapa.html”.

Parámetros

request – El objeto `HttpRequest` que contiene metadatos sobre la solicitud.

Devuelve

Renderiza la plantilla “mapa/mapa.html” con los datos GeoJSON.

Tipo del valor devuelto

`HttpResponse`

8.3 Vista Colectivos

`apps.mapas.viewscollectivos.filter_cueradiocolectivo(request)`

Filtra colectivos y establecimientos de salud cercanos según criterios del usuario.

Esta vista maneja solicitudes POST que contienen criterios para filtrar los colectivos y establecimientos de salud. Realiza una consulta a la base de datos, filtra los resultados basándose en los criterios proporcionados y, si se especifica un radio, busca líneas de colectivos cercanas. Los resultados se devuelven a la plantilla “`mapa/cueradiocolectivos.html`”.

Parámetros

request – El objeto `HttpRequest` que contiene metadatos sobre la solicitud.

Devuelve

Renderiza la plantilla “`mapa/cueradiocolectivos.html`” con los resultados filtrados. Si ocurre un error, renderiza “`error.html`” con un mensaje correspondiente.

Tipo del valor devuelto

`HttpResponse`

`apps.mapas.viewscollectivos.filtrado_cueradiocolectivo(request)`

Renderiza la plantilla del filtro para la búsqueda de colectivos.

Esta vista carga la plantilla HTML “`mapa/filter_colectivos.html`”, donde el usuario puede ingresar criterios de búsqueda para filtrar colectivos.

Parámetros

request – El objeto `HttpRequest` que contiene metadatos sobre la solicitud.

Devuelve

Renderiza la plantilla “`mapa/filter_colectivos.html`”.

Tipo del valor devuelto

`HttpResponse`

8.4 Vista Comisarias

`apps.mapas.viewscomisarias.filter_cueradiocomisarias(request)`

Filtra comisarías y establecimientos cercanos según criterios del usuario.

Esta vista maneja solicitudes POST que contienen criterios para filtrar las comisarías y otros establecimientos cercanos. Realiza una consulta a la base de datos, filtra los resultados basándose en los criterios proporcionados y, si se especifica un radio, busca comisarías cercanas. Los resultados se devuelven a la plantilla “`mapa/cueradiocomisarias.html`”.

Parámetros

request – El objeto `HttpRequest` que contiene metadatos sobre la solicitud.

Devuelve

Renderiza la plantilla “`mapa/cueradiocomisarias.html`” con los resultados filtrados. Si ocurre un error, renderiza “`error.html`” con un mensaje correspondiente.

Tipo del valor devuelto

`HttpResponse`

`apps.mapas.viewscomisarias.filtrado_cueradiocomisarias(request)`

Renderiza la plantilla del filtro para la búsqueda de comisarías.

Esta vista carga la plantilla HTML “`mapa/filter_comisarias.html`”, donde el usuario puede ingresar criterios de búsqueda para filtrar comisarías.

Parámetros

request – El objeto `HttpRequest` que contiene metadatos sobre la solicitud.

Devuelve

Renderiza la plantilla “`mapa/filter_comisarias.html`”.

Tipo del valor devuelto

`HttpResponse`

8.5 Vista Centros Salud

`apps.mapas.viewscentrossalud.filter_cueradiorosalud(request)`

Filtra establecimientos de salud y líneas de colectivos cercanas según criterios del usuario.

Esta vista maneja solicitudes POST que contienen criterios para filtrar los establecimientos de salud. Realiza una consulta a la base de datos, filtra los resultados basándose en los criterios proporcionados y, si se especifica un radio, busca líneas de colectivos cercanas. Los resultados se devuelven a la plantilla “`mapa/cueradiorosalud.html`”.

Parámetros

request – El objeto `HttpRequest` que contiene metadatos sobre la solicitud.

Devuelve

Renderiza la plantilla “`mapa/cueradiorosalud.html`” con los resultados filtrados. Si ocurre un error, renderiza “`error.html`” con un mensaje correspondiente.

Tipo del valor devuelto

`HttpResponse`

`apps.mapas.viewscentrossalud.filtrado_cueradiorosalud(request)`

Renderiza la plantilla del filtro para la búsqueda de establecimientos de salud.

Esta vista carga la plantilla HTML “`mapa/filter_salud.html`”, donde el usuario puede ingresar criterios de búsqueda para filtrar establecimientos de salud.

Parámetros

request – El objeto `HttpRequest` que contiene metadatos sobre la solicitud.

Devuelve

Renderiza la plantilla “`mapa/filter_salud.html`”.

Tipo del valor devuelto

`HttpResponse`

8.6 Vista Cueradio

`apps.mapas.viewscueradio.filter_cueradio(request)`

Filtra los datos de cueras y los devuelve en un formato adecuado para ser renderizados en un mapa.

Parámetros

request (*HttpRequest*) – La solicitud HTTP que se está manejando, esperándose que sea un POST con datos.

Devuelve

La respuesta renderizada con los datos filtrados en “mapa/cueradio.html”, o un JsonResponse en caso de error.

Tipo del valor devuelto

HttpResponse

`apps.mapas.viewscueradio.filtrado_cueradio(request)`

Renderiza la plantilla para el filtrado de datos en el mapa de cueras.

Parámetros

request (*HttpRequest*) – La solicitud HTTP que se está manejando.

Devuelve

La respuesta renderizada con la plantilla “mapa/filter_cuearadio.html”.

Tipo del valor devuelto

HttpResponse

`apps.mapas.viewscueradio.filtrado_establecimiento(request)`

Renderiza la plantilla para el filtrado de establecimientos en el mapa.

Parámetros

request (*HttpRequest*) – La solicitud HTTP que se está manejando.

Devuelve

La respuesta renderizada con la plantilla “mapa/filter_cueradioestablecimiento.html”.

Tipo del valor devuelto

HttpResponse

`apps.mapas.viewscueradio.get_region_data(request)`

Obtiene información sobre la región y sus localidades en base a un “region_pad”.

Parámetros

request (*HttpRequest*) – La solicitud HTTP que se está manejando, debe contener el parámetro “region_pad”.

Devuelve

Un JsonResponse con los datos de la región y localidades, o un error en caso de fallo.

Tipo del valor devuelto

JsonResponse

`apps.mapas.viewscueradio.obtener_geometria(request)`

Obtiene y serializa las geometrías de las regiones en formato GeoJSON.

Parámetros

request (*HttpRequest*) – La solicitud HTTP que se está manejando.

Devuelve

La respuesta renderizada con los datos en “mapa/regionales.html”, o un JsonResponse en caso de error.

Tipo del valor devuelto

HttpResponse

`apps.mapas.viewscueradio.obtener_geometria2(request)`

Obtiene y serializa las geometrías filtradas por “region_pad” en formato GeoJSON.

Parámetros

request (*HttpRequest*) – La solicitud HTTP que se está manejando.

Devuelve

La respuesta renderizada con los datos en “mapa/regionaleselec.html”, o un JsonResponse en caso de error.

Tipo del valor devuelto

HttpResponse

8.7 Formularios

```
class apps.mapas.forms.FiltroFormularioRadio(data=None, files=None, auto_id='id_%s', prefix=None,
                                             initial=None, error_class=class
                                             'django.forms.utils.ErrorList', label_suffix=None,
                                             empty_permitted=False, field_order=None,
                                             use_required_attribute=None, renderer=None)
```

Formulario para filtrar por nivel, localidad y establecimiento.

Atributos:

nivel (ChoiceField): Campo de selección para los niveles, que se rellena dinámicamente. localidad (ChoiceField): Campo de selección para las localidades, que se rellena dinámicamente. establecimiento (ChoiceField): Campo de selección para los establecimientos, que se rellena dinámicamente.

```
base_fields = {'establecimiento': <django.forms.fields.ChoiceField object>,
               'localidad': <django.forms.fields.ChoiceField object>, 'nivel':
               <django.forms.fields.ChoiceField object>}
```

```
declared_fields = {'establecimiento': <django.forms.fields.ChoiceField object>,
                   'localidad': <django.forms.fields.ChoiceField object>, 'nivel':
                   <django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

Documentación de la app Mapoteca

9.1 Modelos

class apps.mapoteca.models.**ArchMapas**(*args, **kwargs)

Bases: Model

Modelo que representa un mapa en la base de datos.

Atributos:

titulo (str): El título del mapa, con un límite de 100 caracteres. Este campo no puede estar en blanco.

archivo (FileField): El archivo del mapa que se subirá a la carpeta “mapoteca/”. ruta (str, optional): La ruta del archivo en el sistema de archivos, puede ser nulo o estar en blanco.

Métodos:

__str__(): Devuelve una representación en forma de cadena del objeto, que es el título del mapa.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

archivo

The descriptor for the file attribute on the model instance. Return a FieldFile when accessed so you can write code like:

```
>>> from myapp.models import MyModel
>>> instance = MyModel.objects.get(pk=1)
>>> instance.file.size
```

Assign a file object on assignment so you can do:

```
>>> with open('/path/to/hello.world') as f:
...     instance.file = File(f)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

ruta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

titulo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

9.2 Vistas

`apps.mapoteca.views.obtener_nombre_mapa(url)`

Extrae el nombre del archivo del mapa a partir de su URL.

Parámetros

url (*str*) – La URL del archivo del mapa.

Devuelve

El nombre del archivo extraído de la URL o “Archivo no disponible” si la URL es vacía.

Tipo del valor devuelto

str

`apps.mapoteca.views.ver_mapas(request)`

Vista para mostrar todos los mapas almacenados en la base de datos.

Obtiene todos los objetos de ArchMapas, extrae el nombre de cada archivo y los pasa a la plantilla para su visualización.

Parámetros

request – El objeto HttpRequest que contiene información sobre la solicitud HTTP.

Devuelve

La respuesta que renderiza la plantilla “mapoteca/ver_mapas.html” con el contexto que incluye la lista de mapas.

Tipo del valor devuelto

HttpResponse

10.1 Modelos

class apps.normativa.models.**ArchNoramtiva**(*args, **kwargs)

Bases: `Model`

Modelo que representa un archivo normativo en la base de datos.

Atributos:

asunto (str): El asunto del archivo normativo. tnorma (str): El tipo de norma asociada al archivo. nro (str): El número de la norma. anio (int): El año de la norma. archivo (FileField): El archivo normativo que se sube. ruta (str): La ruta del archivo en el sistema de archivos, si es aplicable.

Métodos:

save(args, kwargs): Sobrescribe el método de guardado para manejar la apertura y el almacenamiento del archivo de manera adecuada.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

archivo

The descriptor for the file attribute on the model instance. Return a `FieldFile` when accessed so you can write code like:

```
>>> from myapp.models import MyModel
>>> instance = MyModel.objects.get(pk=1)
>>> instance.file.size
```

Assign a file object on assignment so you can do:

```
>>> with open('/path/to/hello.world') as f:
...     instance.file = File(f)
```

asunto

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

año

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nro

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

ruta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(*args, **kwargs)

Sobrescribe el método de guardado para asegurarse de que el archivo se abra correctamente en modo binario antes de guardarlo.

Parámetros

- ***args** – Argumentos posicionales para el método de guardado.
- ****kwargs** – Argumentos nombrados para el método de guardado.

tipo_norma

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

10.2 Vistas

apps.normativa.views.obtener_nombre_archivo(url)

Obtiene el nombre del archivo a partir de su URL.

Parámetros

url (*str*) – La URL del archivo.

Devuelve

El nombre del archivo si la URL es válida, o “Archivo no disponible”.

Tipo del valor devuelto

str

`apps.normativa.views.ver_normas(request)`

Vista que recupera y muestra una lista de normas en la plantilla correspondiente.

Esta vista obtiene todas las instancias del modelo `ArchNormativa`, extrae el nombre del archivo asociado a cada norma y lo añade como un atributo adicional a cada objeto de norma. Luego, renderiza la plantilla “`normativa/ver_normas.html`” con la lista de normas.

Parámetros

request (*HttpRequest*) – El objeto de solicitud HTTP.

Devuelve

La respuesta renderizada de la plantilla con las normas.

Tipo del valor devuelto

`HttpResponse`

11.1 Modelos

class apps.oplectura.models.Periodos(*args, **kwargs)

Bases: Model

Modelo que representa un periodo en el sistema.

periodo

Nombre del periodo.

Type

str

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

periodo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

regevaluacionfluidezlectora_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

```
class apps.oplectura.models.RegDocporSeccion(*args, **kwargs)
```

Bases: Model

Modelo que representa la relación entre docentes y secciones.

dni_docen

DNI del docente.

Type

str

apellido_docen

Apellido del docente.

Type

str

nombres_docen

Nombres del docente.

Type

str

cueanexo

Cueanexo relacionado.

Type

str

curso

Curso al que pertenece el docente.

Type

Curso

division

División al que pertenece el docente.

Type

Division

turno

Turno del docente.

Type

Turno

operativos

Tipo operativo del docente.

Type

TipoOperativo

validacion

Estado de validación del docente.

Type

bool

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

apellido_docen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cueanexo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

curso

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

curso_id**division**

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

division_id**dni_docen**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nombres_docen

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

operativos

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToManyDescriptor instance.

operativos_id

turno

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToManyDescriptor instance.

turno_id

validacion

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class apps.oplectura.models.**RegEvaluacionFluidezLectora**(*args, **kwargs)

Bases: Model

Modelo que representa la evaluación de fluidez lectora de un alumno.

asistencia

Estado de asistencia del alumno.

Type
bool

cueanexo

Cueanexo relacionado.

Type
str

region

Región del alumno.

Type
str

grado

Grado del alumno.

Type
str

seccion

Sección del alumno.

Type

str

tramo

Tramo al que pertenece la evaluación.

Type

Periodos

dni_alumno

DNI del alumno.

Type

str

apellido_alumno

Apellido del alumno.

Type

str

nombres_alumno

Nombres del alumno.

Type

str

velocidad

Velocidad de lectura del alumno.

Type

int

cal_vel

Calificación de la velocidad.

Type

str

precision

Precisión de lectura del alumno.

Type

int

cal_pres

Calificación de la precisión.

Type

str

prosodia

Prosodia del alumno.

Type

int

cal_pros

Calificación de la prosodia.

Type

str

comprension

Comprensión del alumno.

Type

int

cal_comp

Calificación de la comprensión.

Type

str

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

apellido_alumno

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

asistencia

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cal_comp

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cal_pres

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cal_pros

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cal_vel

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

comprension

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cueanexo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

dni_alumno

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_calificacion_comp(valor)

Obtiene la calificación para la comprensión.

Parámetros

valor (*int*) – Valor de comprensión.

Devuelve

Calificación correspondiente a la comprensión.

Tipo del valor devuelto

str

get_calificacion_pres(valor)

Obtiene la calificación para la precisión de lectura.

Parámetros

valor (*int*) – Valor de precisión.

Devuelve

Calificación correspondiente a la precisión.

Tipo del valor devuelto

str

get_calificacion_pros(valor)

Obtiene la calificación para la prosodia.

Parámetros

valor (*int*) – Valor de prosodia.

Devuelve

Calificación correspondiente a la prosodia.

Tipo del valor devuelto

str

get_calificacion_vel(valor)

Obtiene la calificación para la velocidad de lectura.

Parámetros

valor (*int*) – Valor de velocidad.

Devuelve

Calificación correspondiente a la velocidad.

Tipo del valor devuelto

str

grado

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nombres_alumno

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

precision

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

prosodia

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

region

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(*args, **kwargs)

Guarda la instancia del modelo después de calcular las calificaciones.

Parámetros

- ***args** – Positional arguments.
- ****kwargs** – Keyword arguments.

seccion

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

tramo

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):  
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

tramo_id**velocidad**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class apps.oplectura.models.TipoOperativo(*args, **kwargs)

Bases: Model

Modelo que representa un tipo operativo en el sistema.

toperativo

Nombre del tipo operativo.

Type

str

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

regdocporseccion_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

toperativo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class apps.oplectura.models.curso(*args, **kwargs)

Bases: Model

Modelo que representa un curso en el sistema.

nom_curso

Nombre del curso.

Type

str

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom_curso

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

regdocporseccion_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

```
class apps.oplectura.models.division(*args, **kwargs)
```

Bases: Model

Modelo que representa una división en el sistema.

nom_division

Nombre de la división.

Type

str

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom_division

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

regdocporseccion_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

```
class apps.oplectura.models.turno(*args, **kwargs)
```

Bases: Model

Modelo que representa un turno en el sistema.

nom_turno

Nombre del turno.

Type

str

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom_turno

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

regdocporseccion_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

11.2 Vistas

class apps.oplectura.views.CreateRegDocporSeccionView(**kwargs)

Vista para crear un nuevo registro de documento por sección.

Atributos:

model: El modelo RegDocporSeccion. **form_class:** El formulario RegDocporSeccionForm.
template_name: La plantilla utilizada para el formulario. **success_url:** La URL a la que se redirige tras el éxito.

dispatch(request, *args, **kwargs)

form_class

alias de RegDocporSeccionForm

form_valid(form)

Procesa el formulario cuando es válido.

Parámetros

form – El formulario válido.

Devuelve

Redirige a la URL de éxito tras guardar el registro.

Tipo del valor devuelto

HttpResponse

model

alias de RegDocporSeccion

success_url = '/operativo/cargar/'

```
template_name = 'oplectura/regdocporseccion.html'
```

```
apps.oplectura.views.DepEvaluacionPortada(request)
```

Renderiza la plantilla de la portada de evaluación.

Parámetros

request – El objeto de solicitud HTTP.

Devuelve

La respuesta renderizada con la plantilla de portada.

Tipo del valor devuelto

HttpResponse

```
class apps.oplectura.views.EditarAlumnosDirectoresView(**kwargs)
```

form_class

alias de RegEvaluacionFluidezLectoraDirectoresForm

form_valid(form)

If the form is valid, save the associated model.

get_context_data(kwargs)**

Insert the form into the context dict.

get_object()

Return the object the view is displaying.

Require *self.queryset* and a *pk* or *slug* argument in the URLconf. Subclasses can override this to return any object.

model

alias de RegEvaluacionFluidezLectora

success_url = '/operativo/list_alumnos/'

template_name = 'oplectura/editaralumnosdirectores.html'

```
class apps.oplectura.views.EditarDocentesView(**kwargs)
```

Vista para editar un registro de docente.

Atributos:

model: El modelo RegDocporSeccion. form_class: El formulario RegDocporSeccionEdicionForm.

template_name: La plantilla utilizada para editar. success_url: La URL a la que se redirige tras el éxito.

form_class

alias de RegDocporSeccionEdicionForm

form_valid(form)

Procesa el formulario cuando es válido.

Parámetros

form – El formulario válido.

Devuelve

Redirige a la URL de éxito tras la actualización.

Tipo del valor devuelto

HttpResponse

get_context_data(kwargs)**

Agrega contexto adicional a la plantilla.

Parámetros

****kwargs** – Contexto adicional.

Devuelve

Contexto actualizado con el título.

Tipo del valor devuelto

dict

get_object()

Obtiene el objeto a editar.

Devuelve

El objeto correspondiente al ID proporcionado.

Tipo del valor devuelto

RegDocporSeccion

model

alias de RegDocporSeccion

success_url = '/operativo/listados/'

template_name = 'oplectura/editardocentes.html'

class apps.oplectura.views.EditarEvaluacionAlumnosView(**kwargs)

Vista para editar una evaluación de alumnos.

Atributos:

model: El modelo RegEvaluacionFluidezLectora. **form_class:** El formulario RegEvaluacionFluidezLectoraForm. **template_name:** La plantilla utilizada para editar.

form_class

alias de RegEvaluacionFluidezLectoraForm

form_valid(form)

Procesa el formulario cuando es válido.

Parámetros

form – El formulario válido.

Devuelve

Redirige a la URL de éxito tras la actualización.

Tipo del valor devuelto

HttpResponse

get_context_data(kwargs)**

Agrega contexto adicional a la plantilla.

Parámetros

****kwargs** – Contexto adicional.

Devuelve

Contexto actualizado con el título.

Tipo del valor devuelto

dict

get_object()

Obtiene el objeto a editar.

Devuelve

El objeto correspondiente al ID proporcionado.

Tipo del valor devuelto

RegEvaluacionFluidezLectora

get_success_url()

Construye la URL de éxito tras la actualización.

Devuelve

La URL con los parámetros de búsqueda.

Tipo del valor devuelto

str

model

alias de RegEvaluacionFluidezLectora

template_name = 'oplectura/editarevaluacionalumnos.html'

class apps.oplectura.views.EliminarDocentesView(kwargs)**

Vista para eliminar un registro de docente.

Métodos:

get: Elimina el registro de docente y redirige al listado.

get(request)

Elimina el docente especificado.

Parámetros

request – El objeto de solicitud HTTP.

Devuelve

Redirige al listado de docentes.

Tipo del valor devuelto

HttpResponse

class apps.oplectura.views.EliminarEvaluacionAlumnoDirectoresView(kwargs)**

get(request)

class apps.oplectura.views.EliminarEvaluacionAlumnoView(kwargs)**

Vista para eliminar evaluaciones de alumnos.

model

Modelo de RegEvaluacionFluidezLectora.

get(request)

Elimina un registro de evaluación de alumno basado en el ID proporcionado.

Parámetros

request – La solicitud HTTP recibida.

Devuelve

Redirige a la lista de evaluaciones tras la eliminación.

Tipo del valor devuelto

HttpResponse

```

class apps.oplectura.views.ListadoAlumnosDirectoresView(**kwargs)

    context_object_name = 'evaluacionlectora'

    get_context_data(**kwargs)
        Get the context for this view.

    get_queryset()
        Return the list of items for this view.

        The return value must be an iterable and may be an instance of QuerySet in which case QuerySet specific
        behavior will be enabled.

    model
        alias de RegEvaluacionFluidezLectora

    template_name = 'oplectura/listadoalumnosevaluacion.html'

class apps.oplectura.views.ListadoDocentesView(**kwargs)
    Vista para listar los docentes registrados.

    Atributos:
        model: El modelo RegDocporSeccion. template_name: La plantilla utilizada para mostrar el listado.
        context_object_name: El nombre del contexto para los docentes.

    context_object_name = 'docentesporseccion'

    dispatch(request, *args, **kwargs)

    get_context_data(**kwargs)
        Agrega contexto adicional a la plantilla.

        Parámetros
            **kwargs – Contexto adicional.

        Devuelve
            Contexto actualizado con el título.

        Tipo del valor devuelto
            dict

    get_queryset()
        Obtiene la consulta de los docentes.

        Devuelve
            Todos los registros de RegDocporSeccion.

        Tipo del valor devuelto
            QuerySet

    model
        alias de RegDocporSeccion

    template_name = 'oplectura/listadodocentes.html'

class apps.oplectura.views.ListadoEvaluacionLectora(**kwargs)
    Vista para listar las evaluaciones de fluidez lectora.

    Atributos:
        model: El modelo RegEvaluacionFluidezLectora. template_name: La plantilla utilizada para mostrar el
        listado. context_object_name: El nombre del contexto para las evaluaciones.

```

```
context_object_name = 'evaluacionlectora'
```

```
get_context_data(**kwargs)
```

Agrega contexto adicional a la plantilla.

Parámetros

****kwargs** – Contexto adicional.

Devuelve

Contexto actualizado con el título y el formulario de filtro.

Tipo del valor devuelto

dict

```
get_queryset()
```

Filtra las evaluaciones según los parámetros de búsqueda.

Devuelve

Las evaluaciones filtradas.

Tipo del valor devuelto

QuerySet

model

alias de RegEvaluacionFluidezLectora

```
template_name = 'oplectura/listadoevaluacionlectora.html'
```

```
class apps.oplectura.views.ListadoEvaluacionLectoraDirectoresView(**kwargs)
```

```
context_object_name = 'evaluaciondirector'
```

```
get_context_data(**kwargs)
```

Get the context for this view.

```
get_queryset()
```

Return the list of items for this view.

The return value must be an iterable and may be an instance of *QuerySet* in which case *QuerySet* specific behavior will be enabled.

model

alias de RegEvaluacionFluidezLectora

```
template_name = 'oplectura/listadoevaluacionlectoradirector.html'
```

```
class apps.oplectura.views.RegAlumnosFluidezLectoraCreateView(**kwargs)
```

form_class

alias de RegAlumnosFluidezLectoraForm

```
get_context_data(**kwargs)
```

Insert the form into the context dict.

```
get_success_url()
```

Return the URL to redirect to after processing a valid form.

model

alias de RegEvaluacionFluidezLectora

```

    template_name = 'oplectura/regalumnosfluidezlectora_form.html'

class apps.oplectura.views.RegAlumnosFluidezLectoraDirectorCreateView(**kwargs)

    form_class
        alias de RegAlumnosFluidezLectoraDirectorForm

    get_context_data(**kwargs)
        Insert the form into the context dict.

    model
        alias de RegEvaluacionFluidezLectora

    success_url = '/operativo/list_alumnos/'

    template_name = 'oplectura/regalumnosfluidezlectoradirector_form.html'

```

apps.oplectura.views.RegistrarEvaluacionLectora(*request*)

Vista para registrar una evaluación de fluidez lectora.

Parámetros

request – El objeto de solicitud HTTP.

Devuelve

La respuesta renderizada con el formulario o redirige tras el registro.

Tipo del valor devuelto

HttpResponse

apps.oplectura.views_resultados.calcular_estadisticas_por_cueanexo(*username*)

Calcula estadísticas sobre los alumnos por cueanexo.

Parámetros

username (*str*) – El nombre de usuario para filtrar los datos.

Devuelve

Una tupla que contiene:

- resultados_total_dni: Lista de estadísticas por cueanexo y desempeño.
- total_dni_sin_desempenio: Total de DNI sin discriminar por desempeño.
- resultado_promedio_velocidad: Promedio de velocidad.
- resultado_promedio_precision: Promedio de precisión.
- resultado_promedio_prosodia: Promedio de prosodia.
- resultado_promedio_comprension: Promedio de comprensión.
- total_dni_presentes: Total de DNI de alumnos presentes.

Tipo del valor devuelto

tuple

apps.oplectura.views_resultados.cargar_grafico_reg(*request*)

apps.oplectura.views_resultados.mostrar_grafico_reg(*request*)

Muestra el gráfico de evaluación según las regiones, ámbitos y sectores seleccionados.

Parámetros

request (*HttpRequest*) – La solicitud HTTP.

Devuelve

Renderiza la plantilla con los gráficos filtrados.

Tipo del valor devuelto

HttpResponse

```
apps.oplectura.views_resultados.tu_vista(request)
```

Vista que muestra las estadísticas de evaluación lectora.

Parámetros

request (*HttpRequest*) – La solicitud HTTP.

Devuelve

Renderiza la plantilla con gráficos y estadísticas.

Tipo del valor devuelto

HttpResponse

11.3 Formularios

```
class apps.oplectura.forms.FiltroEvaluacionForm(*args, **kwargs)
```

Formulario de filtro para evaluaciones de fluidez lectora.

Fields:

- cueanexo: Selección del cueanexo.
- grado: Selección del grado.
- seccion: Selección de la sección.

```
base_fields = {'cueanexo': <django.forms.fields.ChoiceField object>, 'grado':  
<django.forms.fields.ChoiceField object>, 'seccion':  
<django.forms.fields.ChoiceField object>}
```

```
declared_fields = {'cueanexo': <django.forms.fields.ChoiceField object>, 'grado':  
<django.forms.fields.ChoiceField object>, 'seccion':  
<django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

```
class apps.oplectura.forms.RegAlumnosFluidezLectoraDirectorForm(*args, **kwargs)
```

```
CURSO_CHOICES = [('PRIMERO', 'PRIMERO'), ('SEGUNDO', 'SEGUNDO'), ('TERCERO',  
'TERCERO'), ('CUARTO', 'CUARTO'), ('QUINTO', 'QUINTO'), ('SEXTO', 'SEXTO'),  
('SEPTIMO', 'SEPTIMO')]
```

class Meta

```
fields = '__all__'
```

model

alias de RegEvaluacionFluidezLectora


```

widgets = {'asistencia': <django.forms.widgets.HiddenInput object>, 'cal_comp':
<django.forms.widgets.HiddenInput object>, 'cal_pres':
<django.forms.widgets.HiddenInput object>, 'cal_pros':
<django.forms.widgets.HiddenInput object>, 'cal_vel':
<django.forms.widgets.HiddenInput object>, 'comprension':
<django.forms.widgets.HiddenInput object>, 'precision':
<django.forms.widgets.HiddenInput object>, 'prosodia':
<django.forms.widgets.HiddenInput object>, 'velocidad':
<django.forms.widgets.HiddenInput object>}

REGIONES_CHOICES = [('R.E. 1', 'R.E. 1'), ('R.E. 2', 'R.E. 2'), ('R.E. 3', 'R.E.
3'), ('R.E. 4-A', 'R.E. 4-A'), ('R.E. 4-B', 'R.E. 4-B'), ('R.E. 5', 'R.E. 5'),
('R.E. 6', 'R.E. 6'), ('R.E. 7', 'R.E. 7'), ('R.E. 8-A', 'R.E. 8-A'), ('R.E. 8-B',
'R.E. 8-B'), ('R.E. 9', 'R.E. 9'), ('R.E. 10-A', 'R.E. 10-A'), ('R.E. 10-B', 'R.E.
10-B'), ('R.E. 10-C', 'R.E. 10-C'), ('SUB. R.E. 1-A', 'SUB. R.E. 1-A'), ('SUB. R.E.
1-B', 'SUB. R.E. 1-B'), ('SUB. R.E. 2', 'SUB. R.E. 2'), ('SUB. R.E. 3', 'SUB. R.E.
3'), ('SUB. R.E. 5', 'SUB. R.E. 5')]

SECCION_CHOICES = [('A', 'A'), ('B', 'B'), ('C', 'C'), ('D', 'D'), ('E', 'E'), ('F',
'F'), ('G', 'G'), ('H', 'H'), ('I', 'I'), ('J', 'J'), ('K', 'K'), ('L', 'L'), ('M',
'M'), ('N', 'N'), ('O', 'O'), ('P', 'P'), ('Q', 'Q'), ('R', 'R'), ('S', 'S'), ('T',
'T'), ('U', 'U'), ('V', 'V'), ('W', 'W'), ('X', 'X'), ('Y', 'Y'), ('Z', 'Z'),
('MULTIPLE', 'MULTIPLE')]

base_fields = {'apellido_alumno': <django.forms.fields.CharField object>,
'asistencia': <django.forms.fields.BooleanField object>, 'cal_comp':
<django.forms.fields.CharField object>, 'cal_pres': <django.forms.fields.CharField
object>, 'cal_pros': <django.forms.fields.CharField object>, 'cal_vel':
<django.forms.fields.CharField object>, 'comprension':
<django.forms.fields.IntegerField object>, 'cueanexo':
<django.forms.fields.CharField object>, 'dni_alumno':
<django.forms.fields.CharField object>, 'grado': <django.forms.fields.ChoiceField
object>, 'nombres_alumno': <django.forms.fields.CharField object>, 'precision':
<django.forms.fields.IntegerField object>, 'prosodia':
<django.forms.fields.IntegerField object>, 'region':
<django.forms.fields.ChoiceField object>, 'seccion':
<django.forms.fields.ChoiceField object>, 'tramo':
<django.forms.models.ModelChoiceField object>, 'velocidad':
<django.forms.fields.IntegerField object>}

clean_apellido_alumno()

clean_cueanexo()

clean_dni_alumno()

clean_nombres_alumno()

declared_fields = {'grado': <django.forms.fields.ChoiceField object>, 'region':
<django.forms.fields.ChoiceField object>, 'seccion':
<django.forms.fields.ChoiceField object>}

property media
    Return all media required to render the widgets on this form.

```

```
class apps.oplectura.forms.RegAlumnosFluidezLectoraForm(*args, **kwargs)
```

Formulario para cargar alumnos en evaluaciones de fluidez lectora.

Fields:

- region: Selección de la región (opcional).
- grado: Selección del grado (opcional).
- seccion: Selección de la sección (opcional).
- cal_vel: Calificación de velocidad (oculta).
- cal_pres: Calificación de precisión (oculta).
- cal_pros: Calificación de prosodia (oculta).
- cal_comp: Calificación de comprensión (oculta).
- asistencia: Indicador de asistencia (checkbox).

```
CURSO_CHOICES = [('PRIMERO', 'PRIMERO'), ('SEGUNDO', 'SEGUNDO'), ('TERCERO', 'TERCERO'), ('CUARTO', 'CUARTO'), ('QUINTO', 'QUINTO'), ('SEXTO', 'SEXTO'), ('SEPTIMO', 'SEPTIMO')]
```

```
class Meta
```

```
    fields = '__all__'
```

```
    model
```

```
        alias de RegEvaluacionFluidezLectora
```

```
    widgets = {'asistencia': <django.forms.widgets.CheckboxInput object>,
               'cal_comp': <django.forms.widgets.HiddenInput object>, 'cal_pres':
               <django.forms.widgets.HiddenInput object>, 'cal_pros':
               <django.forms.widgets.HiddenInput object>, 'cal_vel':
               <django.forms.widgets.HiddenInput object>}
```

```
REGIONES_CHOICES = [('R.E. 1', 'R.E. 1'), ('R.E. 2', 'R.E. 2'), ('R.E. 3', 'R.E. 3'), ('R.E. 4-A', 'R.E. 4-A'), ('R.E. 4-B', 'R.E. 4-B'), ('R.E. 5', 'R.E. 5'), ('R.E. 6', 'R.E. 6'), ('R.E. 7', 'R.E. 7'), ('R.E. 8-A', 'R.E. 8-A'), ('R.E. 8-B', 'R.E. 8-B'), ('R.E. 9', 'R.E. 9'), ('R.E. 10-A', 'R.E. 10-A'), ('R.E. 10-B', 'R.E. 10-B'), ('R.E. 10-C', 'R.E. 10-C'), ('SUB. R.E. 1-A', 'SUB. R.E. 1-A'), ('SUB. R.E. 1-B', 'SUB. R.E. 1-B'), ('SUB. R.E. 2', 'SUB. R.E. 2'), ('SUB. R.E. 3', 'SUB. R.E. 3'), ('SUB. R.E. 5', 'SUB. R.E. 5')]
```

```
SECCION_CHOICES = [('A', 'A'), ('B', 'B'), ('C', 'C'), ('D', 'D'), ('E', 'E'), ('F', 'F'), ('G', 'G'), ('H', 'H'), ('I', 'I'), ('J', 'J'), ('K', 'K'), ('L', 'L'), ('M', 'M'), ('N', 'N'), ('O', 'O'), ('P', 'P'), ('Q', 'Q'), ('R', 'R'), ('S', 'S'), ('T', 'T'), ('U', 'U'), ('V', 'V'), ('W', 'W'), ('X', 'X'), ('Y', 'Y'), ('Z', 'Z'), ('MULTIPLE', 'MULTIPLE')]
```

```

base_fields = {'apellido_alumno': <django.forms.fields.CharField object>,
'asistencia': <django.forms.fields.BooleanField object>, 'cal_comp':
<django.forms.fields.CharField object>, 'cal_pres': <django.forms.fields.CharField
object>, 'cal_pros': <django.forms.fields.CharField object>, 'cal_vel':
<django.forms.fields.CharField object>, 'comprension':
<django.forms.fields.IntegerField object>, 'cueanexo':
<django.forms.fields.CharField object>, 'dni_alumno':
<django.forms.fields.CharField object>, 'grado': <django.forms.fields.ChoiceField
object>, 'nombres_alumno': <django.forms.fields.CharField object>, 'precision':
<django.forms.fields.IntegerField object>, 'prosodia':
<django.forms.fields.IntegerField object>, 'region':
<django.forms.fields.ChoiceField object>, 'seccion':
<django.forms.fields.ChoiceField object>, 'tramo':
<django.forms.models.ModelChoiceField object>, 'velocidad':
<django.forms.fields.IntegerField object>}

```

```
clean_apellido_alumno()
```

```
clean_cueanexo()
```

```
clean_dni_alumno()
```

```
clean_nombres_alumno()
```

```

declared_fields = {'grado': <django.forms.fields.ChoiceField object>, 'region':
<django.forms.fields.ChoiceField object>, 'seccion':
<django.forms.fields.ChoiceField object>}

```

property media

Return all media required to render the widgets on this form.

```

class apps.oplectura.forms.RegDocporSeccionEdicionForm(data=None, files=None, auto_id='id_%s',
prefix=None, initial=None,
error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None, empty_permitted=False,
instance=None,
use_required_attribute=None,
renderer=None)

```

Formulario para editar documentos por sección.

Fields son los mismos que RegDocporSeccionForm, excluyendo “id”.

class Meta

```

fields = ['dni_docen', 'apellido_docen', 'nombres_docen', 'cueanexo', 'curso',
'division', 'turno', 'operativos', 'validacion']

```

model

alias de RegDocporSeccion

```
base_fields = {'apellido_docen': <django.forms.fields.CharField object>,
'cueanexo': <django.forms.fields.CharField object>, 'curso':
<django.forms.models.ModelChoiceField object>, 'division':
<django.forms.models.ModelChoiceField object>, 'dni_docen':
<django.forms.fields.CharField object>, 'nombres_docen':
<django.forms.fields.CharField object>, 'operativos':
<django.forms.models.ModelChoiceField object>, 'turno':
<django.forms.models.ModelChoiceField object>, 'validacion':
<django.forms.fields.BooleanField object>}
```

clean_cueanexo()

Valida que el campo “cueanexo” sea un número de 9 dígitos que comience con “22”.

Muestra

ValidationError – Si el “cueanexo” no cumple con el formato requerido.

declared_fields = {}

property media

Return all media required to render the widgets on this form.

```
class apps.oplectura.forms.RegDocporSeccionForm(data=None, files=None, auto_id='id_%s',
prefix=None, initial=None, error_class=<class
'django.forms.utils.ErrorList'>, label_suffix=None,
empty_permitted=False, instance=None,
use_required_attribute=None, renderer=None)
```

Formulario para registrar documentos por sección.

Fields:

- id: Identificador del registro.
- dni_docen: DNI del docente.
- apellido_docen: Apellido del docente.
- nombres_docen: Nombres del docente.
- cueanexo: Código único de anexos (debe comenzar con “22”).
- curso: Curso al que pertenece el documento.
- division: División del curso.
- turno: Turno del curso.
- operativos: Información adicional sobre operativos.
- validacion: Estado de validación del registro.

class Meta

```
fields = ['id', 'dni_docen', 'apellido_docen', 'nombres_docen', 'cueanexo',
'curso', 'division', 'turno', 'operativos', 'validacion']
```

model

alias de RegDocporSeccion

```

base_fields = {'apellido_docen': <django.forms.fields.CharField object>,
'cueanexo': <django.forms.fields.CharField object>, 'curso':
<django.forms.models.ModelChoiceField object>, 'division':
<django.forms.models.ModelChoiceField object>, 'dni_docen':
<django.forms.fields.CharField object>, 'nombres_docen':
<django.forms.fields.CharField object>, 'operativos':
<django.forms.models.ModelChoiceField object>, 'turno':
<django.forms.models.ModelChoiceField object>, 'validacion':
<django.forms.fields.BooleanField object>}

```

```
clean_cueanexo()
```

Valida que el campo “cueanexo” sea un número de 9 dígitos que comience con “22”.

Muestra

ValidationError – Si el “cueanexo” no cumple con el formato requerido.

```
declared_fields = {}
```

property media

Return all media required to render the widgets on this form.

```
class apps.oplectura.forms.RegEvaluacionFluidezLectoraDirectoresForm(*args, **kwargs)
```

```

CURSO_CHOICES = [('PRIMERO', 'PRIMERO'), ('SEGUNDO', 'SEGUNDO'), ('TERCERO',
'TERCERO'), ('CUARTO', 'CUARTO'), ('QUINTO', 'QUINTO'), ('SEXTO', 'SEXTO'),
('SEPTIMO', 'SEPTIMO')]

```

```
class Meta
```

```
fields = '__all__'
```

```
model
```

alias de RegEvaluacionFluidezLectora

```

widgets = {'asistencia': <django.forms.widgets.HiddenInput object>, 'cal_comp':
<django.forms.widgets.HiddenInput object>, 'cal_pres':
<django.forms.widgets.HiddenInput object>, 'cal_pros':
<django.forms.widgets.HiddenInput object>, 'cal_vel':
<django.forms.widgets.HiddenInput object>, 'comprension':
<django.forms.widgets.HiddenInput object>, 'cueanexo':
<django.forms.widgets.HiddenInput object>, 'precision':
<django.forms.widgets.HiddenInput object>, 'prosodia':
<django.forms.widgets.HiddenInput object>, 'region':
<django.forms.widgets.HiddenInput object>, 'velocidad':
<django.forms.widgets.HiddenInput object>}

```

```

SECCION_CHOICES = [('A', 'A'), ('B', 'B'), ('C', 'C'), ('D', 'D'), ('E', 'E'), ('F',
'F'), ('G', 'G'), ('H', 'H'), ('I', 'I'), ('J', 'J'), ('K', 'K'), ('L', 'L'), ('M',
'M'), ('N', 'N'), ('O', 'O'), ('P', 'P'), ('Q', 'Q'), ('R', 'R'), ('S', 'S'), ('T',
'T'), ('U', 'U'), ('V', 'V'), ('W', 'W'), ('X', 'X'), ('Y', 'Y'), ('Z', 'Z'),
('MULTIPLE', 'MULTIPLE')]

```

```
base_fields = {'apellido_alumno': <django.forms.fields.CharField object>,
'asistencia': <django.forms.fields.BooleanField object>, 'cal_comp':
<django.forms.fields.CharField object>, 'cal_pres': <django.forms.fields.CharField
object>, 'cal_pros': <django.forms.fields.CharField object>, 'cal_vel':
<django.forms.fields.CharField object>, 'comprension':
<django.forms.fields.IntegerField object>, 'cueanexo':
<django.forms.fields.CharField object>, 'dni_alumno':
<django.forms.fields.CharField object>, 'grado': <django.forms.fields.ChoiceField
object>, 'nombres_alumno': <django.forms.fields.CharField object>, 'precision':
<django.forms.fields.IntegerField object>, 'prosodia':
<django.forms.fields.IntegerField object>, 'region': <django.forms.fields.CharField
object>, 'seccion': <django.forms.fields.ChoiceField object>, 'tramo':
<django.forms.models.ModelChoiceField object>, 'velocidad':
<django.forms.fields.IntegerField object>}
```

```
clean_apellido_alumno()
```

```
clean_nombres_alumno()
```

```
declared_fields = {'apellido_alumno': <django.forms.fields.CharField object>,
'dni_alumno': <django.forms.fields.CharField object>, 'grado':
<django.forms.fields.ChoiceField object>, 'nombres_alumno':
<django.forms.fields.CharField object>, 'seccion': <django.forms.fields.ChoiceField
object>}
```

property media

Return all media required to render the widgets on this form.

```
class apps.oplectura.forms.RegEvaluacionFluidezLectoraForm(*args, **kwargs)
```

Formulario para cargar evaluaciones de fluidez lectora.

Fields:

- cueanexo: Código único de anexos (oculto).
- region: Región del estudiante (oculta).
- grado: Grado del estudiante (oculto).
- seccion: Sección del estudiante (oculta).
- apellido_alumno: Apellido del alumno (oculto).
- nombres_alumno: Nombres del alumno (oculto).
- dni_alumno: DNI del alumno (oculto).
- cal_vel: Calificación de velocidad (oculta).
- cal_pres: Calificación de precisión (oculta).
- cal_pros: Calificación de prosodia (oculta).
- cal_comp: Calificación de comprensión (oculta).
- asistencia: Indicador de asistencia (checkbox).

class Meta

```
fields = '__all__'
```

model

alias de RegEvaluacionFluidezLectora

```
widgets = {'apellido_alumno': <django.forms.widgets.HiddenInput object>,
'asistencia': <django.forms.widgets.CheckboxInput object>, 'cal_comp':
<django.forms.widgets.HiddenInput object>, 'cal_pres':
<django.forms.widgets.HiddenInput object>, 'cal_pros':
<django.forms.widgets.HiddenInput object>, 'cal_vel':
<django.forms.widgets.HiddenInput object>, 'cueanexo':
<django.forms.widgets.HiddenInput object>, 'dni_alumno':
<django.forms.widgets.HiddenInput object>, 'grado':
<django.forms.widgets.HiddenInput object>, 'nombres_alumno':
<django.forms.widgets.HiddenInput object>, 'region':
<django.forms.widgets.HiddenInput object>, 'seccion':
<django.forms.widgets.HiddenInput object>}
```

```
base_fields = {'apellido_alumno': <django.forms.fields.CharField object>,
'asistencia': <django.forms.fields.BooleanField object>, 'cal_comp':
<django.forms.fields.CharField object>, 'cal_pres': <django.forms.fields.CharField
object>, 'cal_pros': <django.forms.fields.CharField object>, 'cal_vel':
<django.forms.fields.CharField object>, 'compresion':
<django.forms.fields.IntegerField object>, 'cueanexo':
<django.forms.fields.CharField object>, 'dni_alumno':
<django.forms.fields.CharField object>, 'grado': <django.forms.fields.CharField
object>, 'nombres_alumno': <django.forms.fields.CharField object>, 'precision':
<django.forms.fields.IntegerField object>, 'prosodia':
<django.forms.fields.IntegerField object>, 'region': <django.forms.fields.CharField
object>, 'seccion': <django.forms.fields.CharField object>, 'tramo':
<django.forms.models.ModelChoiceField object>, 'velocidad':
<django.forms.fields.IntegerField object>}
```

clean()

Valida los campos del formulario al enviar.

Se asegura de que las calificaciones sean mayores a 0 si la asistencia está marcada.

Muestra

ValidationError – Si las calificaciones no cumplen con las condiciones.

```
declared_fields = {}
```

property media

Return all media required to render the widgets on this form.

12.1 Modelos

12.2 Vistas

`apps.reportes.views.conectar_bd(request)`

`apps.reportes.views.filter_data_cargos(request)`

Vista para procesar los datos del formulario de filtrado de cargos.

Parámetros

request – El objeto de solicitud de Django.

Devuelve

Renderiza la plantilla con los resultados del filtrado
o una plantilla de error si no se encuentra información.

Tipo del valor devuelto

HttpResponse

`apps.reportes.views.filter_data_docentes(request)`

Vista asíncrona para procesar los datos del formulario de filtrado de docentes.

Parámetros

request – El objeto de solicitud de Django.

Devuelve

Renderiza la plantilla con los resultados del filtrado
o una plantilla de error si no se encuentra información.

Tipo del valor devuelto

HttpResponse

`apps.reportes.views.filter_data_horas(request)`

Procesa los datos del formulario de filtrado de horas.

Parámetros

request – Objeto HttpRequest.

Devuelve

Renderiza la plantilla “reportes/horas.html” con los datos filtrados o una plantilla de error si hay problemas con la conexión o la consulta.

`apps.reportes.views.filtrado_cargos(request)`

Vista para mostrar el formulario de filtrado de cargos.

Parámetros

request – El objeto de solicitud de Django.

Devuelve

Renderiza la plantilla de filtrado de cargos.

Tipo del valor devuelto

HttpResponse

`apps.reportes.views.filtrado_docentes(request)`

Vista para mostrar el formulario de filtrado de docentes.

Parámetros

request – El objeto de solicitud de Django.

Devuelve

Renderiza la plantilla de filtrado de docentes.

Tipo del valor devuelto

HttpResponse

`apps.reportes.views.filtrado_horas(request)`

Vista para mostrar el formulario de filtrado de horas.

Parámetros

request – El objeto de solicitud de Django.

Devuelve

Renderiza la plantilla de filtrado de horas.

Tipo del valor devuelto

HttpResponse

`apps.reportes.views_matric.conectar_bd(request)`

Establece una conexión a la base de datos PostgreSQL.

Parámetros

request – El objeto de solicitud HTTP.

Devuelve

Un objeto de conexión a la base de datos si la conexión es exitosa, de lo contrario None.

Tipo del valor devuelto

connection

`apps.reportes.views_matric.filter_data_aborigen(request)`

Procesa los datos del formulario de filtrado de matrícula aborigen y devuelve los resultados.

Parámetros

request – El objeto de solicitud HTTP.

Devuelve

Renderizado de la plantilla correspondiente con los datos obtenidos o un error si no hay datos.

`apps.reportes.views_matric.filter_data_comesp(request)`

Procesa los datos del formulario de filtrado de matrícula común y especial y devuelve los resultados.

Parámetros

request – El objeto de solicitud HTTP.

Devuelve

Renderizado de la plantilla correspondiente con los datos obtenidos o un error si no hay datos.

`apps.reportes.views_matric.filter_data_snu(request)`

Procesa los datos del formulario de filtrado de matrícula SNU y devuelve los resultados.

Parámetros

request – El objeto de solicitud HTTP.

Devuelve

Renderizado de la plantilla correspondiente con los datos obtenidos o un error si no hay datos.

`apps.reportes.views_matric.filtrado_aborigen(request)`

Renderiza el formulario de filtrado de matrícula aborigen.

Parámetros

request – El objeto de solicitud HTTP.

Devuelve

Renderizado de la plantilla “reportes/filter_aborigen.html”.

`apps.reportes.views_matric.filtrado_comesp(request)`

Renderiza el formulario de filtrado de matrícula común y especial.

Parámetros

request – El objeto de solicitud HTTP.

Devuelve

Renderizado de la plantilla “reportes/filter_comesp.html”.

`apps.reportes.views_matric.filtrado_snu(request)`

Renderiza el formulario de filtrado de matrícula SNU.

Parámetros

request – El objeto de solicitud HTTP.

Devuelve

Renderizado de la plantilla “reportes/filter_snu.html”.

`apps.reportes.views_listados.consulta_ofertas(request)`

Consulta las ofertas educativas por departamento.

Esta vista obtiene los departamentos únicos y realiza una consulta SQL para contar las ofertas educativas según diferentes criterios. Si se selecciona un departamento, se aplicará un filtro en la consulta.

Parámetros

request – La solicitud HTTP que contiene los parámetros de consulta.

Devuelve

Si la solicitud es AJAX, devuelve los datos en formato JSON. HttpResponse: Renderiza la plantilla “reportes/listadosestablecimientos.html” con los departamentos y datos obtenidos.

Tipo del valor devuelto

JsonResponse

`apps.reportes.views_listados.consulta_ofertas_reg(request)`

Consulta las ofertas educativas por región.

Esta vista obtiene las regiones únicas y las localidades asociadas, y realiza una consulta SQL para contar las ofertas educativas según diferentes criterios. Si se selecciona una región, se aplicará un filtro en la consulta.

Parámetros

request – La solicitud HTTP que contiene los parámetros de consulta.

Devuelve

Si la solicitud es AJAX, devuelve los datos en formato JSON. `HttpResponse`: Renderiza la plantilla “reportes/listadosestablecreg.html” con las regiones y datos obtenidos.

Tipo del valor devuelto

`JsonResponse`

`apps.reportes.views_infograf.equipoview(request)`

Muestra la información del equipo.

Esta vista renderiza la plantilla “equipo.html”, que contiene información sobre el equipo.

Parámetros

request – La solicitud HTTP.

Devuelve

Renderiza la plantilla del equipo.

Tipo del valor devuelto

`HttpResponse`

`apps.reportes.views_infograf.infografiaview(request)`

Muestra la infografía de matrícula, cargos y horas.

Esta vista renderiza la plantilla “reportes/infografia.html”, que contiene información sobre la matrícula, los cargos y las horas.

Parámetros

request – La solicitud HTTP.

Devuelve

Renderiza la plantilla infografía con el contexto adecuado.

Tipo del valor devuelto

`HttpResponse`

`apps.reportes.views_infograf.infografiaview2(request)`

Muestra una segunda infografía.

Esta vista renderiza la plantilla “reportes/infografia2.html”, que contiene información adicional en formato de infografía.

Parámetros

request – La solicitud HTTP.

Devuelve

Renderiza la segunda plantilla de infografía.

Tipo del valor devuelto

`HttpResponse`

`apps.reportes.views_infodocatividad.consulta_docentes_actividad(request)`

Consulta los resultados de docentes por actividad y departamento.

Obtiene una lista de departamentos únicos y realiza una consulta SQL para obtener resultados relacionados con la actividad de los docentes en un departamento específico.

Parámetros

request – La solicitud HTTP que puede contener el departamento seleccionado.

Devuelve

Si la solicitud es AJAX, devuelve un JsonResponse con los resultados.

Si es una solicitud estándar, renderiza la plantilla con los resultados, los departamentos, los colores y el total general.

Tipo del valor devuelto

HttpResponse

`apps.reportes.views_carrerastitulos.consulta_carrerastitulos(request)`

Consulta los títulos de carreras y localidades en función de los filtros seleccionados.

Parámetros

request – La solicitud HTTP que contiene los filtros seleccionados.

Devuelve

Si la solicitud es AJAX, devuelve un JsonResponse con los títulos y datos filtrados.

Si es una solicitud estándar, renderiza la plantilla con las localidades, niveles, títulos y datos.

Tipo del valor devuelto

HttpResponse

`apps.reportes.views_carrerastitulos.datoscarreras(request)`

Obtiene los datos de una carrera específica basándose en el cueanexo proporcionado.

Parámetros

request – La solicitud HTTP que contiene el cueanexo.

Devuelve

Renderiza la plantilla del modal con los resultados obtenidos de la base de datos.

Si no se proporciona cueanexo o hay un error de conexión, renderiza una página de error.

Tipo del valor devuelto

HttpResponse

13.1 Modelos

class apps.supervisores.models.DirectoresRegionales(*args, **kwargs)

Bases: Model

Modelo que representa a un Director Regional.

Atributos:

dni_reg (str): El número de identificación único del director. apellido_reg (str): El apellido del director. nombres_reg (str): El/los nombre(s) del director. email_reg (str): La dirección de correo electrónico del director. telefono_reg (str): El número de teléfono del director. region_reg (str): La designación regional del director.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

apellido_reg

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

dni_reg

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

email_reg

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nombres_reg

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

region_reg

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

telefono_reg

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class apps.supervisores.models.EscuelaSupervisor(*args, **kwargs)

Bases: Model

Modelo que representa una escuela supervisada por un Supervisor.

Atributos:

cueanexo (str): El identificador único de la escuela. region_esc (str): La designación regional de la escuela. oferta (str): La oferta educativa proporcionada por la escuela. modalidad (str): La modalidad de educación (por ejemplo, común, técnica). supervisor (Supervisor): El Supervisor que supervisa la escuela.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

cueanexo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

modalidad

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

oferta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

region_esc

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

supervisor

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:


```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

supervisor_id

```
class apps.supervisores.models.Supervisor(*args, **kwargs)
```

Bases: Model

Modelo que representa a un Supervisor.

Atributos:

dni (str): El número de identificación único del supervisor. apellido (str): El apellido del supervisor. nombres (str): El/los nombre(s) del supervisor. email (str): La dirección de correo electrónico del supervisor. telefono (str): El número de teléfono del supervisor. region (str): La designación regional del supervisor.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

apellido

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

dni

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

escuelas

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nombres

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

region

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

telefono

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

13.2 Vistas

```
class apps.supervisores.views.DirectorRegionalCreateView(**kwargs)
```

Vista para crear un nuevo Director Regional.

Contexto:

- form: Formulario para crear un Director Regional.

```
fields = '__all__'
```

model

alias de DirectoresRegionales

```
success_url = '/super/directores/'
```

```
template_name = 'supervisores/crear_director_regional.html'
```

```
class apps.supervisores.views.DirectorRegionalDeleteView(**kwargs)
```

Vista para eliminar un Director Regional.

Redirige a la lista de Directores Regionales tras la eliminación.

```
get(request)
```

```
class apps.supervisores.views.DirectorRegionalListView(**kwargs)
```

Vista para listar todos los Directores Regionales.

Contexto:

- directores: Lista de Directores Regionales.
- title: Título de la vista.

```
context_object_name = 'directores'
```

```
get_context_data(**kwargs)
```

Get the context for this view.

```
get_queryset()
```

Return the list of items for this view.

The return value must be an iterable and may be an instance of *QuerySet* in which case *QuerySet* specific behavior will be enabled.

model

alias de DirectoresRegionales

```
template_name = 'supervisores/lista_directores_regionales.html'
```

```
class apps.supervisores.views.DirectorRegionalUpdateView(**kwargs)
```

Vista para editar un Director Regional existente.

Contexto:

- form: Formulario para editar un Director Regional.

```
fields = '__all__'
```

model

alias de DirectoresRegionales

```
success_url = '/super/directores/'
```

```
template_name = 'supervisores/crear_director_regional.html'
```

```
class apps.supervisores.views.EscuelaCreateView(**kwargs)
```

Vista para crear una nueva Escuela Supervisada.

Contexto:

- form: Formulario para crear una Escuela.

form_class

alias de EscuelaForm

model

alias de EscuelaSupervisor

```
success_url = '/super/escuelas/'
```

```
template_name = 'supervisores/crear_escuela.html'
```

```
class apps.supervisores.views.EscuelaListView(**kwargs)
```

Vista para listar todas las Escuelas Supervisadas.

Contexto:

- escuelas: Lista de Escuelas filtradas por región.
- form: Formulario de filtro por región.
- title: Título de la vista.

```
context_object_name = 'escuelas'
```

```
get_context_data(**kwargs)
```

Añade el formulario de filtro y el título al contexto.

```
get_queryset()
```

Retorna un queryset filtrado por región si se proporciona.

model

alias de EscuelaSupervisor

```
template_name = 'supervisores/lista_escuelas.html'
```

```
class apps.supervisores.views.EscuelaUpdateView(**kwargs)
```

Vista para editar una Escuela Supervisada existente.

Contexto:

- form: Formulario para editar una Escuela.

- title: Título de la vista.

form_class

alias de EscuelaForm

form_valid(form)

If the form is valid, save the associated model.

get_context_data(kwargs)**

Insert the form into the context dict.

get_object()

Return the object the view is displaying.

Require *self.queryset* and a *pk* or *slug* argument in the URLconf. Subclasses can override this to return any object.

model

alias de EscuelaSupervisor

success_url = '/super/escuelas/'

template_name = 'supervisores/crear_escuela.html'

class apps.supervisores.views.EscuelasDeleteView(**kwargs)

Vista para eliminar una Escuela Supervisada.

Redirige a la lista de Escuelas tras la eliminación.

get(request)

class apps.supervisores.views.SupervisorCreateView(**kwargs)

Vista para crear un nuevo Supervisor.

Contexto:

- form: Formulario para crear un Supervisor.

form_class

alias de SupervisorForm

model

alias de Supervisor

success_url = '/super/supervisores/'

template_name = 'supervisores/crear_supervisor.html'

class apps.supervisores.views.SupervisorDeleteView(**kwargs)

Vista para eliminar un Supervisor.

Redirige a la lista de Supervisores tras la eliminación.

get(request)

Elimina el Supervisor basado en el ID proporcionado y redirige a la lista.

class apps.supervisores.views.SupervisorListView(**kwargs)

Vista para listar todos los Supervisores.

Contexto:

- supervisores: Lista de Supervisores filtrados por región.

- form: Formulario de filtro por región.
- title: Título de la vista.

context_object_name = 'supervisores'

get_context_data(**kwargs)

Get the context for this view.

get_queryset()

Return the list of items for this view.

The return value must be an iterable and may be an instance of *QuerySet* in which case *QuerySet* specific behavior will be enabled.

model

alias de Supervisor

template_name = 'supervisores/lista_supervisores.html'

class apps.supervisores.views.**SupervisorUpdateView**(**kwargs)

Vista para editar un Supervisor existente.

Contexto:

- form: Formulario para editar un Supervisor.
- title: Título de la vista.

form_class

alias de SupervisorForm

form_valid(form)

If the form is valid, save the associated model.

get_context_data(**kwargs)

Insert the form into the context dict.

get_object()

Return the object the view is displaying.

Require *self.queryset* and a *pk* or *slug* argument in the URLconf. Subclasses can override this to return any object.

model

alias de Supervisor

success_url = '/super/supervisores/'

template_name = 'supervisores/crear_supervisor.html'

13.3 Formularios

```
class apps.supervisores.forms.EscuelaForm(data=None, files=None, auto_id='id_%s', prefix=None,  
                                          initial=None, error_class=<class  
                                          'django.forms.utils.ErrorList'>, label_suffix=None,  
                                          empty_permitted=False, instance=None,  
                                          use_required_attribute=None, renderer=None)
```

Formulario para crear o editar un objeto EscuelaSupervisor.

region_esc

Campo para seleccionar la región de la escuela.

Type

ChoiceField

oferta

Campo para seleccionar la oferta educativa.

Type

ChoiceField

modalidad

Campo para seleccionar la modalidad educativa.

Type

ChoiceField

Meta:

model (EscuelaSupervisor): El modelo asociado a este formulario. fields (list): Todos los campos del modelo EscuelaSupervisor.

```
MODALIDADES_CHOICES = [('COMÚN', 'COMÚN'), ('TÉCNICO PROFESIONAL', 'TÉCNICO  
PROFESIONAL'), ('ESPECIAL', 'ESPECIAL'), ('JÓVENES Y ADULTOS', 'JÓVENES Y ADULTOS'),  
('ARTÍSTICA', 'ARTÍSTICA'), ('RURAL', 'RURAL'), ('BILINGÜE INTERCULTURAL', 'BILINGÜE  
INTERCULTURAL'), ('CONTEXTO DE ENCIERRO', 'CONTEXTO DE ENCIERRO'), ('HOSPITALARIA -  
DOMICILIARIA', 'HOSPITALARIA - DOMICILIARIA'), ('EDUCACIÓN FÍSICA', 'EDUCACIÓN  
FÍSICA')]
```

class Meta

```
fields = '__all__'
```

model

alias de EscuelaSupervisor

```
OFERTAS_CHOICES = [('INICIAL', 'INICIAL'), ('PRIMARIO', 'PRIMARIO'), ('SECUNDARIO',  
'SECUNDARIO'), ('SUPERIOR', 'SUPERIOR'), ('SERVICIOS EDUCATIVOS', 'SERVICIOS  
EDUCATIVOS')]
```

```
REGIONES_CHOICES = [('R.E. 1', 'R.E. 1'), ('R.E. 2', 'R.E. 2'), ('R.E. 3', 'R.E.  
3'), ('R.E. 4-A', 'R.E. 4-A'), ('R.E. 4-B', 'R.E. 4-B'), ('R.E. 5', 'R.E. 5'),  
('R.E. 6', 'R.E. 6'), ('R.E. 7', 'R.E. 7'), ('R.E. 8-A', 'R.E. 8-A'), ('R.E. 8-B',  
'R.E. 8-B'), ('R.E. 9', 'R.E. 9'), ('R.E. 10-A', 'R.E. 10-A'), ('R.E. 10-B', 'R.E.  
10-B'), ('R.E. 10-C', 'R.E. 10-C'), ('SUB. R.E. 1-A', 'SUB. R.E. 1-A'), ('SUB. R.E.  
1-B', 'SUB. R.E. 1-B'), ('SUB. R.E. 2', 'SUB. R.E. 2'), ('SUB. R.E. 3', 'SUB. R.E.  
3'), ('SUB. R.E. 5', 'SUB. R.E. 5')]
```

```
base_fields = {'cueanexo': <django.forms.fields.CharField object>, 'modalidad':
<django.forms.fields.ChoiceField object>, 'oferta':
<django.forms.fields.ChoiceField object>, 'region_esc':
<django.forms.fields.ChoiceField object>, 'supervisor':
<django.forms.models.ModelChoiceField object>}
```

clean_cueanexo()

Valida el campo “cueanexo”.

Asegura que el Cueanexo comience con “22” y contenga 9 dígitos en total.

Devuelve

El cueanexo validado.

Tipo del valor devuelto

str

Muestra

ValidationError – Si el cueanexo no cumple con los criterios.

```
declared_fields = {'modalidad': <django.forms.fields.ChoiceField object>, 'oferta':
<django.forms.fields.ChoiceField object>, 'region_esc':
<django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

class apps.supervisores.forms.**FiltroRegionalForm**(*args, **kwargs)

Formulario para filtrar resultados por región.

region

Campo opcional para seleccionar la región.

Type

ChoiceField

```
base_fields = {'region': <django.forms.fields.ChoiceField object>}
```

```
declared_fields = {'region': <django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

```
class apps.supervisores.forms.SupervisorForm(data=None, files=None, auto_id='id_%s', prefix=None,
initial=None, error_class=<class
'django.forms.utils.ErrorList'>, label_suffix=None,
empty_permitted=False, instance=None,
use_required_attribute=None, renderer=None)
```

Formulario para crear o editar un objeto Supervisor.

region

Campo para seleccionar la región del supervisor.

Type

ChoiceField

Meta:

model (Supervisor): El modelo asociado a este formulario. fields (list): Todos los campos del modelo Supervisor.

```
class Meta

    fields = '__all__'

    model
        alias de Supervisor

REGIONES_CHOICES = [('R.E. 1', 'R.E. 1'), ('R.E. 2', 'R.E. 2'), ('R.E. 3', 'R.E. 3'), ('R.E. 4-A', 'R.E. 4-A'), ('R.E. 4-B', 'R.E. 4-B'), ('R.E. 5', 'R.E. 5'), ('R.E. 6', 'R.E. 6'), ('R.E. 7', 'R.E. 7'), ('R.E. 8-A', 'R.E. 8-A'), ('R.E. 8-B', 'R.E. 8-B'), ('R.E. 9', 'R.E. 9'), ('R.E. 10-A', 'R.E. 10-A'), ('R.E. 10-B', 'R.E. 10-B'), ('R.E. 10-C', 'R.E. 10-C'), ('SUB. R.E. 1-A', 'SUB. R.E. 1-A'), ('SUB. R.E. 1-B', 'SUB. R.E. 1-B'), ('SUB. R.E. 2', 'SUB. R.E. 2'), ('SUB. R.E. 3', 'SUB. R.E. 3'), ('SUB. R.E. 5', 'SUB. R.E. 5')]

base_fields = {'apellido': <django.forms.fields.CharField object>, 'dni': <django.forms.fields.CharField object>, 'email': <django.forms.fields.EmailField object>, 'nombres': <django.forms.fields.CharField object>, 'region': <django.forms.fields.ChoiceField object>, 'telefono': <django.forms.fields.CharField object>}

clean_apellido()
    Valida el campo “apellido”.

    Asegura que el apellido contenga solo letras en mayúsculas, con tildes, apóstrofes y espacios.

    Devuelve
        El apellido validado en mayúsculas.

    Tipo del valor devuelto
        str

    Muestra
        ValidationError – Si el apellido no cumple con los criterios.

clean_dni()
    Valida el campo “dni”.

    Asegura que el DNI contenga solo números y tenga al menos 7 dígitos.

    Devuelve
        El DNI validado.

    Tipo del valor devuelto
        str

    Muestra
        ValidationError – Si el DNI no cumple con los criterios.

clean_nombres()
    Valida el campo “nombres”.

    Asegura que los nombres contengan solo letras en mayúsculas, con tildes, apóstrofes y espacios.

    Devuelve
        Los nombres validados en mayúsculas.

    Tipo del valor devuelto
        str

    Muestra
        ValidationError – Si los nombres no cumplen con los criterios.
```



```
declared_fields = {'region': <django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

14.1 Modelos

class apps.usuarios.models.CustomUserManager(*args, **kwargs)

Bases: BaseUserManager

Administrador de usuarios personalizado para gestionar la creación de usuarios.

Métodos:

create_user: Crea y devuelve un usuario normal. create_superuser: Crea y devuelve un superusuario.

create_superuser(username, password=None, **extra_fields)

Crea y devuelve un superusuario con un nombre de usuario y una contraseña.

Parámetros

- **username** (*str*) – El nombre de usuario para el superusuario.
- **password** (*str*, *optional*) – La contraseña del superusuario.
- ****extra_fields** – Campos adicionales a añadir al superusuario.

Muestra

ValueError – Si el campo is_staff o is_superuser no está establecido a True.

Devuelve

El superusuario creado.

Tipo del valor devuelto

UsuariosVisualizador

create_user(username, password=None, **extra_fields)

Crea y devuelve un usuario con un nombre de usuario y una contraseña.

Parámetros

- **username** (*str*) – El nombre de usuario para el nuevo usuario.

- **password** (*str*, *optional*) – La contraseña del nuevo usuario.
- ****extra_fields** – Campos adicionales a añadir al usuario.

Muestra

ValueError – Si el campo de nombre de usuario no está definido.

Devuelve

El usuario creado.

Tipo del valor devuelto

UsuariosVisualizador

```
class apps.usuarios.models.NivelAcceso(*args, **kwargs)
```

Bases: Model

Modelo que representa los niveles de acceso para los usuarios.

Atributos:

tacceso (*str*): El tipo de acceso, debe ser único.

exception DoesNotExist

Bases: ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: MultipleObjectsReturned

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

tacceso

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

usuariosvisualizador_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

```
class apps.usuarios.models.UsuariosVisualizador(*args, **kwargs)
```

Bases: AbstractBaseUser, PermissionsMixin

Modelo que representa a los usuarios visualizadores en el sistema.

Atributos:

username (*str*): Nombre de usuario único. apellido (*str*): Apellido del usuario. nombres (*str*): Nombres del usuario. correo (*str*): Correo electrónico del usuario. telefono (*str*): Teléfono del usuario. nivelacceso (NivelAcceso): Nivel de acceso del usuario, con relación a NivelAcceso. activo (*bool*): Indica si el usuario está activo. is_staff (*bool*): Indica si el usuario es parte del personal. is_superuser (*bool*): Indica si el usuario tiene privilegios de superusuario.

Métodos:

`__str__`: Retorna el nombre de usuario.

exception DoesNotExist

Bases: `ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `MultipleObjectsReturned`

`REQUIRED_FIELDS` = ['apellido', 'nombres', 'correo', 'telefono', 'nivelacceso']

`USERNAME_FIELD` = 'username'

activo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

apellido

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

correo

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

groups

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_staff

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_superuser

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

last_login

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

logentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

nivelacceso

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

nivelacceso_id

nombres

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <apps.usuarios.models.CustomUserManager object>

password

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

registroacceso_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

telefono

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

trayectoria_ocupacional

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Place.restaurant is a ReverseOneToOneDescriptor instance.

user_permissions

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

username

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

14.2 Vistas

```
class apps.usuarios.views.EliminarUsuarioView(**kwargs)
```

Vista para eliminar un usuario.

Métodos:

`get`: Elimina el usuario basado en el ID proporcionado.

`get(request)`

Elimina el usuario y redirige a la lista de usuarios.

Parámetros

request – La solicitud HTTP.

Devuelve

Redirección a la lista de usuarios.

Tipo del valor devuelto

`HttpResponseRedirect`

```
class apps.usuarios.views.EliminarUsuarioView_op(**kwargs)
```

Vista para eliminar un usuario en evaluación.

Métodos:

`get`: Elimina el usuario basado en el ID proporcionado.

`get(request)`

Elimina el usuario y redirige a la lista de directores.

Parámetros

request – La solicitud HTTP.

Devuelve

Redirección a la lista de directores.

Tipo del valor devuelto

`HttpResponseRedirect`

```
apps.usuarios.views.check_user_status(request)
```

```
class apps.usuarios.views.crear_usuarios(**kwargs)
```

Vista para crear un nuevo usuario.

Atributos:

model: El modelo a utilizar (UsuariosVisualizador). form_class: El formulario para crear usuarios.
template_name: La plantilla a renderizar. success_url: URL a la que redirigir después de una creación exitosa.

Métodos:

get_context_data: Agrega información adicional al contexto. form_valid: Guarda el usuario en la base de datos.

form_class

alias de UsuariosForm

form_valid(form)

Procesa el formulario válido y guarda el usuario.

Parámetros

form – El formulario enviado.

Devuelve

Respuesta de redirección después de guardar el usuario.

Tipo del valor devuelto

HttpResponse

get_context_data(kwargs)**

Agrega el título al contexto.

Parámetros

****kwargs** – Argumentos adicionales.

Devuelve

El contexto actualizado.

Tipo del valor devuelto

dict

model

alias de UsuariosVisualizador

success_url = '/usua/listado/'

template_name = 'usuarios/crear.html'

```
class apps.usuarios.views.editar_usuarios(**kwargs)
```

Vista para editar un usuario existente.

Atributos:

model: El modelo a utilizar (UsuariosVisualizador). form_class: El formulario para editar usuarios.
template_name: La plantilla a renderizar. success_url: URL a la que redirigir después de una edición exitosa.

Métodos:

get_object: Obtiene el objeto de usuario a editar. get_context_data: Agrega información adicional al contexto. form_valid: Procesa el formulario válido.

form_class

alias de UsuariosForm

form_valid(form)

Procesa el formulario válido y actualiza el usuario.

Parámetros

form – El formulario enviado.

Devuelve

Respuesta de redirección después de actualizar el usuario.

Tipo del valor devuelto

HttpResponse

get_context_data(kwargs)**

Agrega el título al contexto.

Parámetros

****kwargs** – Argumentos adicionales.

Devuelve

El contexto actualizado.

Tipo del valor devuelto

dict

get_object()

Obtiene el objeto de usuario a editar basado en el ID proporcionado.

Devuelve

El usuario a editar.

Tipo del valor devuelto

UsuariosVisualizador

model

alias de UsuariosVisualizador

success_url = '/usua/listado/'

template_name = 'usuarios/editar.html'

class apps.usuarios.views.editar_usuarios_op(kwargs)**

Vista para editar un usuario existente para evaluación.

Atributos:

model: El modelo a utilizar (UsuariosVisualizador). **form_class:** El formulario para editar usuarios.
template_name: La plantilla a renderizar. **success_url:** URL a la que redirigir después de una edición exitosa.

Métodos:

get_object: Obtiene el objeto de usuario a editar. **get_context_data:** Agrega información adicional al contexto. **form_valid:** Procesa el formulario válido.

form_class

alias de UsuariosForm

form_valid(form)

Procesa el formulario válido y actualiza el usuario.

Parámetros

form – El formulario enviado.

Devuelve

Respuesta de redirección después de actualizar el usuario.

Tipo del valor devuelto

HttpResponse

get_context_data(kwargs)**

Agrega el título al contexto.

Parámetros

****kwargs** – Argumentos adicionales.

Devuelve

El contexto actualizado.

Tipo del valor devuelto

dict

get_object()

Obtiene el objeto de usuario a editar basado en el ID proporcionado.

Devuelve

El usuario a editar.

Tipo del valor devuelto

UsuariosVisualizador

model

alias de UsuariosVisualizador

success_url = '/usua/listado_op/'

template_name = 'usuarios/editar_op.html'

class apps.usuarios.views.listado_usuarios(kwargs)**

Vista que muestra el listado de usuarios para el administrador.

Atributos:

model: El modelo a utilizar (UsuariosVisualizador). template_name: La plantilla a renderizar.

context_object_name: Nombre del contexto a pasar a la plantilla.

Métodos:

get_context_data: Agrega información adicional al contexto.

context_object_name = 'usuarios'

get_context_data(kwargs)**

Agrega el título al contexto.

Parámetros

****kwargs** – Argumentos adicionales.

Devuelve

El contexto actualizado.

Tipo del valor devuelto

dict

model

alias de UsuariosVisualizador

```

    template_name = 'usuarios/listado.html'

class apps.usuarios.views.listado_usuarios_op(**kwargs)
    Vista que muestra el listado de usuarios con nivel de acceso «Director/a».

    Atributos:
        model: El modelo a utilizar (UsuariosVisualizador).  template_name: La plantilla a renderizar.
        context_object_name: Nombre del contexto a pasar a la plantilla.

    Métodos:
        get_queryset: Filtra los usuarios por nivel de acceso.  get_context_data: Agrega información adicional al
        contexto.

    context_object_name = 'usuarios'

    get_context_data(**kwargs)
        Agrega el título al contexto.

        Parámetros
            **kwargs – Argumentos adicionales.

        Devuelve
            El contexto actualizado.

        Tipo del valor devuelto
            dict

    get_queryset()
        Filtra los usuarios cuyo nivel de acceso es «Director/a».

        Devuelve
            Los usuarios filtrados.

        Tipo del valor devuelto
            QuerySet

    model
        alias de UsuariosVisualizador

    template_name = 'usuarios/listado_op.html'

class apps.usuarios.views.registrar_usuarios(**kwargs)

    form_class
        alias de UsuariosForm_login

    form_valid(form)
        If the form is valid, save the associated model.

    get_context_data(**kwargs)
        Insert the form into the context dict.

    model
        alias de UsuariosVisualizador

    success_url = '/usua/registro/'

    template_name = 'login/login.html'

```

14.3 Formularios

```
class apps.usuarios.forms.CustomPasswordResetForm(data=None, files=None, auto_id='id_%s',
                                                    prefix=None, initial=None, error_class=<class
                                                    'django.forms.utils.ErrorList'>, label_suffix=None,
                                                    empty_permitted=False, field_order=None,
                                                    use_required_attribute=None, renderer=None)
```

Formulario personalizado para restablecer la contraseña de UsuariosVisualizador.

Este formulario extiende el formulario de restablecimiento de contraseña predeterminado y personaliza el método para obtener usuarios.

Métodos:

`get_users`: Obtiene usuarios activos que pueden ser autenticados.

```
base_fields = {'email': <django.forms.fields.EmailField object>, 'username':
<django.forms.fields.CharField object>}
```

```
declared_fields = {'email': <django.forms.fields.EmailField object>, 'username':
<django.forms.fields.CharField object>}
```

`get_users(username)`

Obtiene una lista de usuarios activos que coinciden con el nombre de usuario proporcionado.

Parámetros:

`username`: El nombre de usuario a buscar.

Retorna:

Un generador de usuarios activos que tienen una contraseña utilizable.

property media

Return all media required to render the widgets on this form.

```
class apps.usuarios.forms.CustomSetPasswordForm(user, *args, **kwargs)
```

Formulario personalizado para establecer una nueva contraseña.

Este formulario extiende el formulario de establecimiento de contraseña y personaliza la validación de las contraseñas.

Métodos:

`clean_new_password2`: Valida que las dos contraseñas coincidan. `save`: Guarda la nueva contraseña establecida.

```
base_fields = {'new_password1': <django.forms.fields.CharField object>,
'new_password2': <django.forms.fields.CharField object>}
```

`clean_new_password2()`

Valida que las dos contraseñas proporcionadas coincidan.

Retorna:

La segunda contraseña limpia.

Lanza:

`ValidationError`: Si las contraseñas no coinciden.

```
declared_fields = {'new_password1': <django.forms.fields.CharField object>,
'new_password2': <django.forms.fields.CharField object>}
```

property media

Return all media required to render the widgets on this form.

save(commit=True)

Guarda la nueva contraseña establecida.

Parámetros:

commit: Si se debe guardar la instancia de usuario en la base de datos.

Retorna:

El objeto de usuario guardado con la nueva contraseña.

```
class apps.usuarios.forms.ResetpassWordForm(data=None, files=None, auto_id='id_%s', prefix=None,
initial=None, error_class=<class
'django.forms.utils.ErrorList'>, label_suffix=None,
empty_permitted=False, field_order=None,
use_required_attribute=None, renderer=None)
```

Formulario para restablecer la contraseña a través del nombre de usuario.

Este formulario permite ingresar un nombre de usuario para iniciar el proceso de restablecimiento.

Métodos:

None.

```
base_fields = {'username': <django.forms.fields.CharField object>}
```

```
declared_fields = {'username': <django.forms.fields.CharField object>}
```

property media

Return all media required to render the widgets on this form.

```
class apps.usuarios.forms.UsuariosForm(*args, **kwargs)
```

Formulario para crear o actualizar instancias del modelo UsuariosVisualizador.

Este formulario incluye validaciones para los campos y encripta la contraseña antes de guardarla.

Métodos:

clean_username: Valida que el nombre de usuario cumpla con los requisitos. save: Guarda la instancia con la contraseña encriptada.

class Meta

```
fields = ['id', 'username', 'password', 'apellido', 'nombres', 'correo',
'telefono', 'nivelacceso', 'activo', 'is_staff']
```

model

alias de UsuariosVisualizador

```
base_fields = {'activo': <django.forms.fields.BooleanField object>, 'apellido':
<django.forms.fields.CharField object>, 'correo': <django.forms.fields.EmailField
object>, 'is_staff': <django.forms.fields.BooleanField object>, 'nivelacceso':
<django.forms.models.ModelChoiceField object>, 'nombres':
<django.forms.fields.CharField object>, 'password': <django.forms.fields.CharField
object>, 'telefono': <django.forms.fields.CharField object>, 'username':
<django.forms.fields.CharField object>}
```

clean_username()

Valida el nombre de usuario para asegurarse de que contenga solo números.

Retorna:

El nombre de usuario limpio.

Lanza:

forms.ValidationError: Si el nombre de usuario no es válido.

declared_fields = {}

property media

Return all media required to render the widgets on this form.

save(commit=True)

Guarda la instancia del formulario con la contraseña encriptada.

Parámetros:

commit: Si se debe guardar la instancia en la base de datos.

Retorna:

La instancia guardada del modelo UsuariosVisualizador.

```
class apps.usuarios.forms.UsuariosForm_login(data=None, files=None, auto_id='id_%s', prefix=None,
initial=None, error_class=<class
'django.forms.utils.ErrorList'>, label_suffix=None,
empty_permitted=False, instance=None,
use_required_attribute=None, renderer=None)
```

Formulario para el inicio de sesión de UsuariosVisualizador.

Este formulario incluye validaciones y widgets personalizados para facilitar la entrada de datos.

Métodos:

None.

class Meta

```
fields = ['username', 'password', 'apellido', 'nombres', 'correo', 'telefono',
'nivelacceso', 'activo', 'is_staff', 'is_superuser']
```

model

alias de UsuariosVisualizador

```
widgets = {'activo': <django.forms.widgets.CheckboxInput object>, 'correo':
<django.forms.widgets.EmailInput object>, 'is_staff':
<django.forms.widgets.CheckboxInput object>, 'is_superuser':
<django.forms.widgets.CheckboxInput object>, 'nivelacceso':
<django.forms.widgets.Select object>, 'telefono':
<django.forms.widgets.TextInput object>}
```

```
base_fields = {'activo': <django.forms.fields.BooleanField object>, 'apellido':
<django.forms.fields.CharField object>, 'correo': <django.forms.fields.EmailField
object>, 'is_staff': <django.forms.fields.BooleanField object>, 'is_superuser':
<django.forms.fields.BooleanField object>, 'nivelacceso':
<django.forms.models.ModelChoiceField object>, 'nombres':
<django.forms.fields.CharField object>, 'password': <django.forms.fields.CharField
object>, 'telefono': <django.forms.fields.CharField object>, 'username':
<django.forms.fields.CharField object>}
```

```
declared_fields = {'apellido': <django.forms.fields.CharField object>, 'nombres':
<django.forms.fields.CharField object>, 'password': <django.forms.fields.CharField
object>, 'username': <django.forms.fields.CharField object>}
```

property media

Return all media required to render the widgets on this form.

`apps.usuarios.forms.validate_alphanumeric_uppercase(value)`

Valida que el valor contenga solo caracteres alfanuméricos y esté en mayúsculas.

Parámetros:

value: El valor a validar.

Lanza:

ValidationError: Si el valor no cumple con las condiciones.

`apps.usuarios.forms.validate_username(value)`

Valida que el nombre de usuario contenga solo números, entre 7 y 9 dígitos.

Parámetros:

value: El valor del nombre de usuario a validar.

Lanza:

ValidationError: Si el valor no cumple con las condiciones.

14.4 Backends

class `apps.usuarios.backends.CustomAuthBackend`

Backend de autenticación personalizado para el modelo UsuariosVisualizador.

Este backend permite la autenticación de usuarios utilizando el modelo UsuariosVisualizador y comparando la contraseña cifrada con la proporcionada.

Métodos:

authenticate: Autentica a un usuario con nombre de usuario y contraseña. get_user: Recupera un usuario basado en su ID. check_password: Verifica si la contraseña proporcionada coincide con la almacenada.

authenticate(*request, username=None, password=None*)

Autentica a un usuario utilizando su nombre de usuario y contraseña.

Parámetros:

request: La solicitud HTTP en curso. username: Nombre de usuario del usuario a autenticar. password: Contraseña proporcionada para la autenticación.

Retorna:

El objeto UsuarioVisualizador si la autenticación es exitosa, de lo contrario None.

check_password(*user, password*)

Verifica si la contraseña proporcionada coincide con la almacenada.

Parámetros:

user: El objeto UsuarioVisualizador para el cual se verifica la contraseña. password: La contraseña proporcionada por el usuario.

Retorna:

True si las contraseñas coinciden, de lo contrario False.

get_user(*user_id*)

Recupera un usuario basado en su ID.

Parámetros:

user_id: El ID del usuario a recuperar.

Retorna:

El objeto UsuarioVisualizador correspondiente al ID, o None si no existe.

14.5 Mixins

class apps.usuarios.mixins.**AdminRequiredMixin**

Mezcla que requiere que el usuario sea parte de un grupo específico para acceder a una vista.

Este mixin extiende UserPassesTestMixin y verifica si el usuario autenticado pertenece al grupo «Administrador». Si no tiene permiso, se lanzará una excepción de permiso denegado.

Métodos:

test_func: Verifica si el usuario pertenece a un grupo permitido. handle_no_permission: Maneja el caso en que el usuario no tiene permisos.

handle_no_permission()

Maneja el caso en que el usuario no tiene permisos para acceder a la vista.

Lanza:

PermissionDenied: Si el usuario no pertenece a ninguno de los grupos permitidos.

test_func()

Verifica si el usuario autenticado pertenece al grupo permitido.

Retorna:

bool: True si el usuario pertenece al grupo “Administrador”, False en caso contrario.

15.1 Vistas

`apps.videoteca.views.videoteca(request)`

Renderiza la página de la videoteca con una lista de videos educativos.

Esta vista proporciona un contexto que incluye un conjunto de videos, cada uno con un título y una URL, que se mostrarán en la plantilla correspondiente.

Parámetros

request (*HttpRequest*) – La solicitud HTTP enviada por el navegador.

Devuelve

La respuesta renderizada que contiene la plantilla “videoteca/videoteca.html” con el contexto de los videos.

Tipo del valor devuelto

HttpResponse

- search