

doc

API Documentation

February 24, 2018

Contents

Contents	1
1 Module __main__	3
1.1 Functions	3
1.2 Variables	3
2 Package model	4
2.1 Modules	4
2.2 Variables	4
3 Module model.Ciudad	5
3.1 Variables	5
3.2 Class Ciudad	5
3.2.1 Methods	5
4 Module modelCurso	6
4.1 Variables	6
4.2 Class Curso	6
4.2.1 Methods	6
5 Module model.Departamento	8
5.1 Variables	8
5.2 Class Departamento	8
5.2.1 Methods	8
6 Module model.Estudiante	9
6.1 Variables	9
6.2 Class Estudiante	9
6.2.1 Methods	9
6.2.2 Properties	10
7 Module model.Persona	11
7.1 Variables	11
7.2 Class Persona	11
7.2.1 Methods	12
7.2.2 Properties	13
8 Module model.Planificacion	14

8.1	Variables	14
8.2	Class Planificacion	14
8.2.1	Methods	14
9	Module model.Profesor	15
9.1	Variables	15
9.2	Class Profesor	15
9.2.1	Methods	16
9.2.2	Properties	17
10	Package view	18
10.1	Variables	18
Index		19

1 Module __main__

1.1 Functions

run()
Metodo de inicializacion

1.2 Variables

Name	Description
__package__	Value: None

2 Package model

2.1 Modules

- **Ciudad** (*Section 3, p. 5*)
- **Curso** (*Section 4, p. 6*)
- **Departamento** (*Section 5, p. 8*)
- **Estudiante** (*Section 6, p. 9*)
- **Persona** (*Section 7, p. 11*)
- **Planificacion** (*Section 8, p. 14*)
- **Profesor** (*Section 9, p. 15*)

2.2 Variables

Name	Description
<code>--package--</code>	Value: None

3 Module model.Ciudad

3.1 Variables

Name	Description
<code>--package--</code>	Value: None

3.2 Class Ciudad

Clase Ciudad, con datos especificos del mismo

3.2.1 Methods

<code>--init--(self, nombre, descripcion, id=0)</code> Constructor de Ciudad Parameters nombre: Nombre de la Ciudad <i>(type=str)</i> descripcion: descripcion de la Ciudad <i>(type=str)</i> id: Identificacion de la Ciudad <i>(type=integer)</i>
<code>--str--(self)</code> Convierte clase a str Return Value Str convertido de la clase <i>(type=str)</i>
<code>set_id(self, id)</code> Modifica id de la ciudad Parameters id: Identificacion nueva <i>(type=integer)</i>
<code>get_nom(self)</code> Retorna Nombre de la Ciudad Return Value Nombre de la Ciudad <i>(type=str)</i>

4 Module model.Curso

4.1 Variables

Name	Description
<code>--package--</code>	Value: 'model'

4.2 Class Curso

Clase Curso

4.2.1 Methods

<code>--init--(self, nombre, aula, tiempo, dia_reunion, edificio, id=0)</code>	
Constructor de clase Curso	
Parameters	
nombre:	Nombre del Curso (<i>type=str</i>)
aula:	Aula donde se va hacer el curso (<i>type=str</i>)
id:	Identificacion del curso (<i>type=integer</i>)
tiempo:	Tiempo en horas del curso (<i>type=double</i>)
dia_reunion:	Fecha de Reunion (<i>type=datetime</i>)
edificio:	Edificio donde es el lugar de Reunion (<i>type=str</i>)
<code>--str--(self)</code>	
Conversion de Clase a str	
Return Value	
Conversion de Clase (<i>type=str</i>)	

set_profesor(*self*, *profesor*)

Pone profesor a cargo del curso

Parameters

profesor: Profesor encargado del curso
(*type=Profesor*)

5 Module model.Departamento

5.1 Variables

Name	Description
<code>--package--</code>	Value: None

5.2 Class Departamento

Clase Departamento

5.2.1 Methods

`--init--(self, nombre, ciudad, telefono, id=0)`

Constructor de clase Departamento

Parameters

nombre: Nombre del Departamento
(*type=str*)

ciudad: Ciudad donde esta el Departamento
(*type=Ciudad*)

telefono: Telefono de contacto del Departamento
(*type=integer*)

id: Identificacion de Departamento
(*type=integer*)

`--str--(self)`

Conversion de clase a str

Return Value

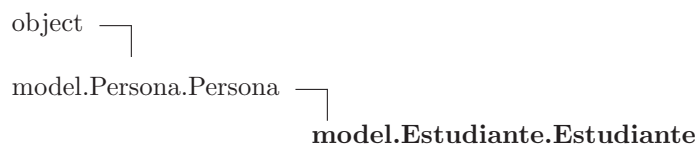
Conversion de clase
(*type=str*)

6 Module *model.Estudiante*

6.1 Variables

Name	Description
<code>--package--</code>	Value: <code>'model'</code>

6.2 Class *Estudiante*



Clase de los estudiantes

6.2.1 Methods

`__init__(self, num, nombre1, nombre2, apellido1, apellido2, edad, lug_nacimiento, lug_residencia, dir_residencia, is_graduado=False)`

Constructor de estudiante, tiene datos de persona, quien es su super-clase

Parameters

`is_graduado`: Dato de si esta graduado o no
(*type=bool*)

Overrides: `object.__init__`

`es_is_graduado(self)`

Retorna en español si esta graduado o no

Return Value

Retorno de graduado en español
(*type=str*)

`__str__(self)`

Conversion de clase a str de Estudiante

Return Value

Clase convertida a str
(*type=str*)

Overrides: `object.__str__`

set_consejero (<i>self</i> , <i>consejero</i>)

Pone consejero a estudiante solo si esta graduado

Parameters

consejero : Profesor consejero que esta disponible (<i>type=Profesor</i>)

Return Value

retrona true si se puso consejero correctamente y false de no se estudiante graduado (<i>type=bool</i>)
--

set_planificacion (<i>self</i> , <i>planificacion</i>)

Modifica y relaciona la planificacion dada
--

Parameters

planificacion : Planificacion que se va a vincular a estudiante (<i>type=Planificacion</i>)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__subclasshook__()`

6.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

7 Module *model.Persona*

7.1 Variables

Name	Description
<code>--package--</code>	Value: None

7.2 Class *Persona*

object └─
 model.Persona.Persona

Known Subclasses: *model.Estudiante.Estudiante*, *model.Profesor.Profesor*

Representacion de los datos basicos de la persona

7.2.1 Methods

`__init__`(*self*, *num*, *nombre1*, *nombre2*, *apellido1*, *apellido2*, *edad*, *lug_nacimiento*, *ciu_residencia*, *dir_residencia*)

Constructor de Persona

Parameters

num:	Numero de identificacion, Unico e Irrepetible (<i>type=integer</i>)
nombre1:	Primer nombre (<i>type=str</i>)
nombre2:	Segundo nombre (<i>type=str</i>)
apellido1:	Primer apellido (<i>type=str</i>)
apellido2:	Segundo apellido (<i>type=str</i>)
edad:	Edad de persona (<i>type=int</i>)
lug_nacimiento:	Lugar de nacimiento (<i>type=Ciudad</i>)
ciu_residencia:	Ciudad de Residencia (<i>type=Ciudad</i>)
dir_residencia:	Direccin de Residencia (<i>type=str</i>)

Overrides: object.__init__

`__str__`(*self*)

Conversion a str de clase

Return Value

Datos de clase en str
(*type=str*)

Overrides: object.__str__

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__subclasshook__()`

7.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

8.1 Variables

Name	Description
--package--	Value: None

Clase de Planificacion de los estudiantes

8.2.1 Methods

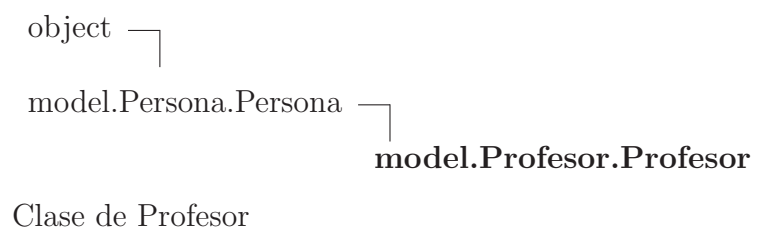
--init-- (<i>self</i> , <i>semestre</i> , <i>id_estudiante</i> , <i>id_curso</i>)	
Constructor de clase de Planificacion	
Parameters	
id_estudiante:	Codigo unico del estudiante (<i>type=integer</i>)
id_curso:	Codigo unico del curso (<i>type=integer</i>)
semestre:	Semestre en el que el estudiante se inscribio (<i>type=integer</i>)
--str-- (<i>self</i>)	
Conversion de clase a str	
Return Value	
Conversion de clase (<i>type=str</i>)	
set_nota (<i>self</i> , <i>nota</i>)	
Poner nota al estudiante que esta inscrito en el curso	
Parameters	
nota:	Nota obtenida del curso (<i>type=double</i>)

9 Module model.Profesor

9.1 Variables

Name	Description
<code>--package--</code>	Value: <code>'model'</code>

9.2 Class Profesor



9.2.1 Methods

```
__init__(self, num, nombre1, nombre2, apellido1, apellido2, edad,
lug_nacimiento, lug_residencia, dir_residencia, titulo, contrato,
is_visitante=False, ini_nombramiento=None, fin_nombramiento=None)
```

Constructor de clase de Profesor

Parameters

is_visitante: Obtener si un profesor es visitante o no
(*type=bool*)

titulo: Titulo obtenido
(*type=str*)

contrato: Contrato que se ha firmado ante la universidad para ejercer enseñanza
(*type=str*)

ini_nombramiento: Fecha de inicio de Nombramiento solo si es profesor visitante
(*type=datetime*)

fin_nombramiento: Fecha de fin de Nombramiento solo si es profesor visitante
(*type=datetime*)

Overrides: object.__init__

```
es_is_visitante(self)
```

Retorna en español si esta visitante o no

Return Value

Retorno de visitante en español
(*type=str*)

```
__str__(self)
```

Conversion de clase a str de Profesor

Return Value

Datos de clase en str
(*type=str*)

Overrides: object.__str__

set_departamento (<i>self</i> , <i>departamento</i>)
Pone el profesor en un departamento
Parameters <i>departamento</i> : Departamento a cambiar (<i>type=Departamento</i>)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__subclasshook__()`

9.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

10 Package view

10.1 Variables

Name	Description
__package__	Value: None

Index

- `--main--` (*module*), 3
 - `--main--run` (*function*), 3
- `model` (*package*), 4
 - `model.Ciudad` (*module*), 5
 - `model.Ciudad.Ciudad` (*class*), 5
 - `model.Curso` (*module*), 6–7
 - `model.Curso.Curso` (*class*), 6–7
 - `model.Departamento` (*module*), 8
 - `model.Departamento.Departamento` (*class*), 8
 - `model.Estudiante` (*module*), 9–10
 - `model.Estudiante.Estudiante` (*class*), 9–10
 - `model.Persona` (*module*), 11–13
 - `model.Persona.Persona` (*class*), 11–13
 - `model.Planificacion` (*module*), 14
 - `model.Planificacion.Planificacion` (*class*), 14
 - `model.Profesor` (*module*), 15–17
 - `model.Profesor.Profesor` (*class*), 15–17
- `view` (*package*), 18