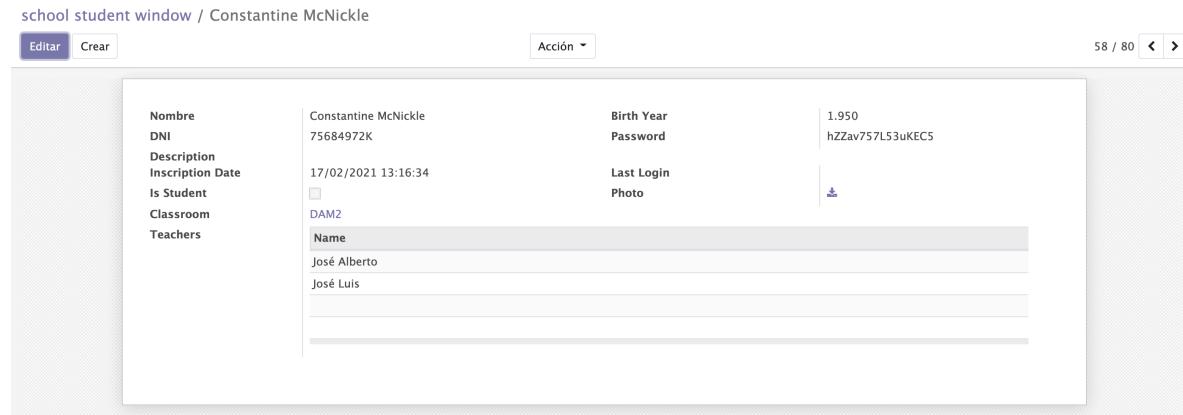


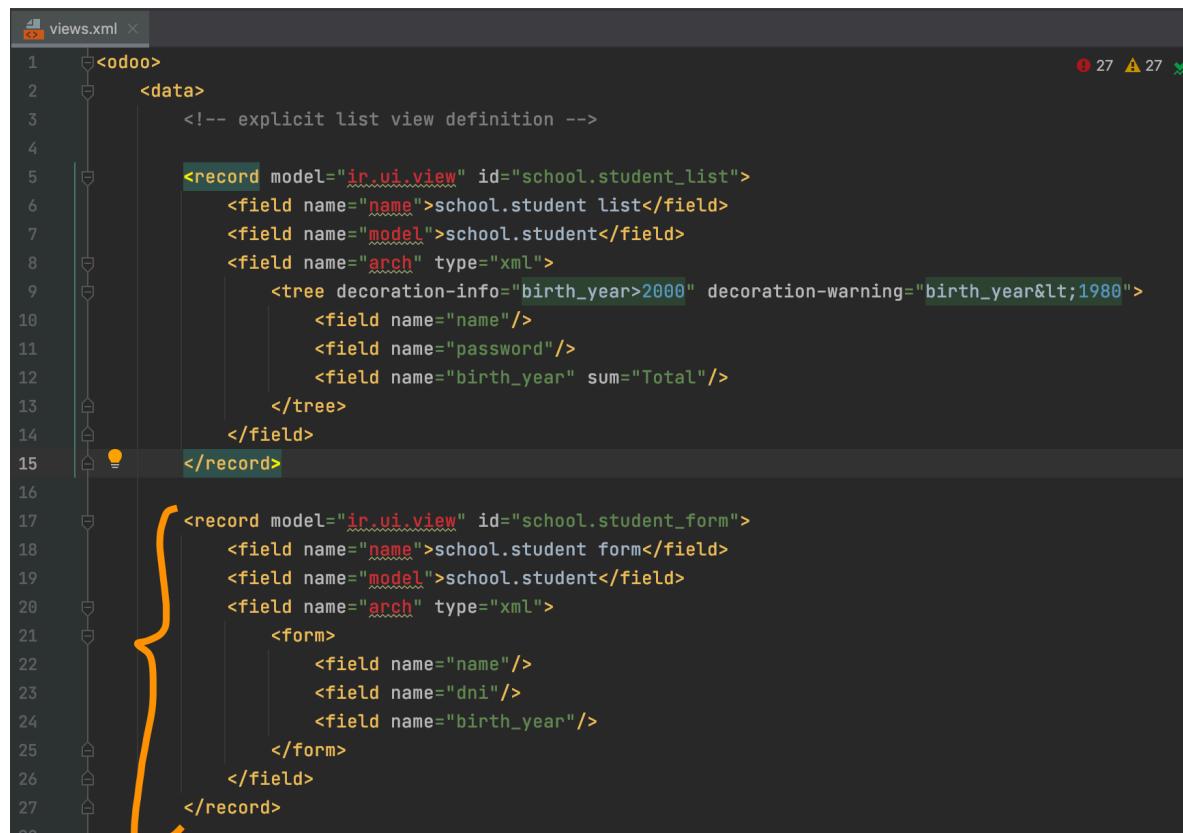
VISTAS FORM

- Vamos a ver las vistas formulario. Las que aparecen por defecto ahora están generaditas automáticamente, por ejemplo:



The screenshot shows a student record in the Odoo interface. The record belongs to Constantine McNickle, with DNI 75684972K, Description 17/02/2021 13:16:34, Is Student checked, Classroom DAM2, and Teachers listed as José Alberto and José Luis. The Birth Year is 1.950 and the Password is hZZav757L53uKECS. There is also a Last Login field and a Photo placeholder.

- Si generamos un registro nosotros con el tipo de vista form, como podemos ver:



```
<odoo>
    <data>
        <!-- explicit list view definition -->

        <record model="ir.ui.view" id="school.student_list">
            <field name="name">school.student list</field>
            <field name="model">school.student</field>
            <field name="arch" type="xml">
                <tree decoration-info="birth_year>2000" decoration-warning="birth_year<1980">
                    <field name="name"/>
                    <field name="password"/>
                    <field name="birth_year" sum="Total"/>
                </tree>
            </field>
        </record>

        <record model="ir.ui.view" id="school.student_form">
            <field name="name">school.student form</field>
            <field name="model">school.student</field>
            <field name="arch" type="xml">
                <form>
                    <field name="name"/>
                    <field name="dni"/>
                    <field name="birth_year"/>
                </form>
            </field>
        </record>
    </data>
</odoo>
```

The code shows two records in the views.xml file. The first record defines a list view for the school.student model, featuring a tree view with fields for name, password, and birth_year, and a warning for birth years between 1980 and 2000. The second record defines a form view for the same model, featuring a form view with fields for name, dni, and birth_year.

- Obtendríamos lo siguiente:

school student window / Fannie Tidbold

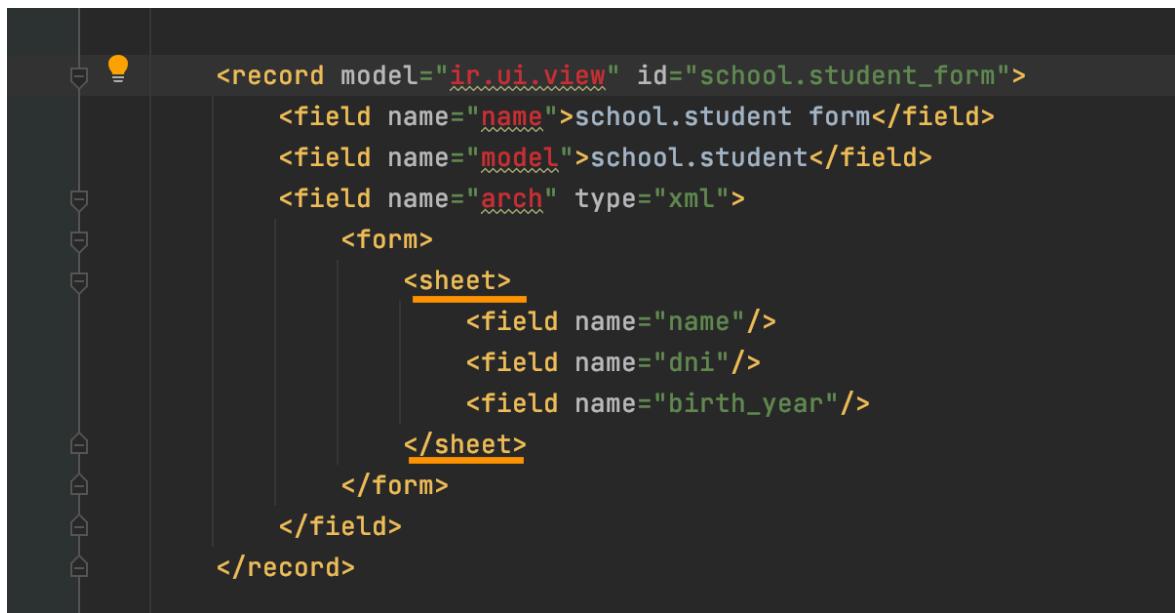
Editar

Crear

Acción ▾

Fannie Tidbold77516174X2.003

- Me dio la información que le pedí pero sin ningún estilo. Antes la generaba Odoo automáticamente, y ahora muestra lo que le especificamos pero de esta forma no se visualiza bien.
- Lo primero que es necesario es la etiqueta **<sheet>** para que asocie el cls correspondiente y no sabrá crear la estructura HTML correctamente. Con esto ya conseguimos cierta estructura;



```
<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <field name="name"/>
                <field name="dni"/>
                <field name="birth_year"/>
            </sheet>
        </form>
    </field>
</record>
```

school student window / Fannie Tidbold

Editar

Crear

Acción ▾

1 / 1 < >

Fannie Tidbold77516174X2.003

- El **<sheet>** necesita que los fields se encuentren dentro de un **<group>**

```

<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <group>
                    <field name="name"/>
                    <field name="dni"/>
                    <field name="birth_year"/>
                </group>
            </sheet>
        </form>
    </field>
</record>

```

school student window / Fannie Tidbold

		Acción	
Editar	Crear		1 / 1 <
Nombre	Fannie Tidbold		
DNI	77516174X		
Birth Year	2.003		

- Si añadimos todos los campos que tenemos:

```

<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <group>
                    <field name="name"/>
                    <field name="birth_year"/>
                    <field name="password"/>
                    <field name="dni"/>
                    <field name="description"/>
                    <field name="inscription_date"/>
                    <field name="last_login"/>
                    <field name="is_student"/>
                    <field name="photo"/>
                    <field name="classroom"/>
                    <field name="teachers"/>
                </group>
            </sheet>
        </form>
    </field>
</record>

```

- Sin hacer nada más, obtenemos:

The screenshot shows the Odoo interface for a student record. At the top, there are buttons for 'Editar' (Edit) and 'Crear' (Create). To the right, there is a dropdown menu labeled 'Acción' and a page number '1 / 1'. The main area displays a table with student information and a list of teachers.

Nombre	Fannie Tidbold			
Birth Year	2.003			
Password	G-O9lrMHFlUv3F7N			
DNI	77516174X			
Description				
Inscription Date	17/02/2021 13:16:28			
Last Login				
Is Student				
Photo				
Classroom	DAM2			
Teachers	<table border="1"> <thead> <tr> <th>Name</th> </tr> </thead> <tbody> <tr> <td>José Alberto</td> </tr> <tr> <td>José Luis</td> </tr> </tbody> </table>	Name	José Alberto	José Luis
Name				
José Alberto				
José Luis				

- El primer problema es que la foto no es visible. Para que lo sea, vamos a cambiar

el orden y vamos a establecerle un widget de tipo imagen.



```
<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <group>
                    <field name="photo" widget="image"/>
                    <field name="name"/>
                    <field name="birth_year"/>
                    <field name="password"/>
                    <field name="dni"/>
                    <field name="description"/>
                    <field name="inscription_date"/>
                    <field name="last_login"/>
                    <field name="is_student"/>
                    <field name="classroom"/>
                    <field name="teachers"/>
                </group>
            </sheet>
        </form>
    </field>
</record>
```

school student window / Fannie Tidbold

Editar Crear Acción ▾ 1 / 1

Photo	Fannie Tidbold			
Nombre	Fannie Tidbold			
Birth Year	2.003			
Password	G-O9lrMHFiUv3F7N			
DNI	77516174X			
Description				
Inscription Date	17/02/2021 13:16:28			
Last Login				
Is Student				
Classroom	DAM2			
Teachers	<table border="1"><thead><tr><th>Name</th></tr></thead><tbody><tr><td>José Alberto</td></tr><tr><td>José Luis</td></tr></tbody></table>	Name	José Alberto	José Luis
Name				
José Alberto				
José Luis				

- Como podemos ver la foto tiene el tamaño que establecimos (200 x 200) en el

modelo.

- Si sacamos la foto fuera del grupo, no aparecerá la etiqueta.

```
<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheets>
                <sheet>
                    <group>
                        <field name="name"/>
                        <field name="birth_year"/>
                        <field name="password"/>
                        <field name="dni"/>
                        <field name="description"/>
                        <field name="inscription_date"/>
                        <field name="last_login"/>
                        <field name="is_student"/>
                        <field name="classroom"/>
                        <field name="teachers"/>
                    </group>
                </sheet>
            </sheets>
        </form>
    </field>
</record>
```

[Editar](#) [Crear](#)[Acción ▾](#)

1 / 1



Nombre	Fannie Tidbold					
Birth Year	2.003					
Password	G-O9lrMHFIUv3F7N					
DNI	77516174X					
Description						
Inscription Date	17/02/2021 13:16:28					
Last Login						
Is Student	<input type="checkbox"/>					
Classroom	DAM2					
Teachers	<table border="1"> <thead> <tr><th>Name</th></tr> </thead> <tbody> <tr><td>José Alberto</td></tr> <tr><td>José Luis</td></tr> <tr><td> </td></tr> <tr><td> </td></tr> </tbody> </table>	Name	José Alberto	José Luis		
Name						
José Alberto						
José Luis						

- Si le doy a editar, tengo la opción de cambiar todo, incluida la foto:

[Guardar](#) [Descartar](#)

1 / 1



Nombre	<input type="text" value="Fannie Tidbold"/>					
Birth Year	<input type="text" value="2.003"/>					
Password	<input type="text" value="G-O9lrMHFIUv3F7N"/>					
DNI	<input type="text" value="77516174X"/>					
Description	<input type="text"/>					
Inscription Date	<input type="text" value="17/02/2021 13:16:28"/>					
Last Login	<input type="text"/>					
Is Student	<input type="checkbox"/>					
Classroom	<input type="text" value="DAM2"/> [x]					
Teachers	<table border="1"> <thead> <tr><th>Name</th></tr> </thead> <tbody> <tr><td>José Alberto</td></tr> <tr><td>José Luis</td></tr> <tr><td> </td></tr> <tr><td> </td></tr> </tbody> </table>	Name	José Alberto	José Luis		
Name						
José Alberto						
José Luis						

- Vamos a ver distintas formas de organizar los campos, en lugar de una lista para que no quede tan largo.
- La primera solución va a ser establecer varios grupos.

```

<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <field name="photo" widget="image"/>
                <group>
                    <group>
                        <field name="name"/>
                        <field name="birth_year"/>
                        <field name="password"/>
                        <field name="dni"/>
                        <field name="description"/>
                    </group>
                    <group>
                        <field name="inscription_date"/>
                        <field name="last_login"/>
                        <field name="is_student"/>
                        <field name="classroom"/>
                    </group>
                </group>
                <group><field name="teachers"/></group>
            </sheet>
        </form>
    </field>
</record>

```

- De esta manera conseguimos lo siguiente:

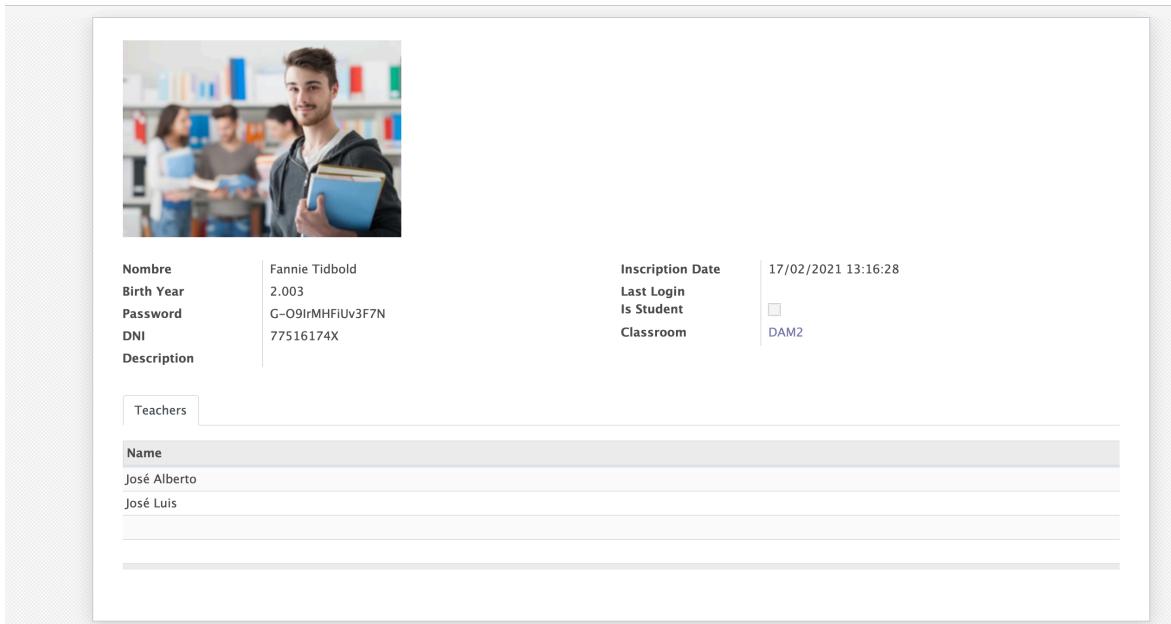
school student window / Fannie Tidbold

Nombre	Fannie Tidbold	Inscription Date	17/02/2021 13:16:28								
Birth Year	2.003	Last Login									
Password	G-O9lrMHFIUv3F7N	Is Student									
DNI	77516174X	Classroom	DAM2								
Description											
Teachers	<table border="1"> <tr> <td>Name</td> <td>José Alberto</td> </tr> <tr> <td></td> <td>José Luis</td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </table>			Name	José Alberto		José Luis				
Name	José Alberto										
	José Luis										

- Hay una etiqueta **<notebook>** y dentro podemos utilizar la etiqueta **<page>**.

Mediante esto se crea una estructura de pestañas. Con esto conseguimos lo siguiente:

```
<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <field name="photo" widget="image"/>
                <group>
                    <group>
                        <field name="name"/>
                        <field name="birth_year"/>
                        <field name="password"/>
                        <field name="dni"/>
                        <field name="description"/>
                    </group>
                    <group>
                        <field name="inscription_date"/>
                        <field name="last_login"/>
                        <field name="is_student"/>
                        <field name="classroom"/>
                    </group>
                </group>
                <notebook>
                    <page string="Teachers">
                        <field name="teachers"/>
                    </page>
                </notebook>
            </sheet>
        </form>
    </field>
</record>
```



- Los profesores son un tree de profesores. Vamos a añadir algunos campos más en el modelo de **teacher**.

```

class teacher(models.Model):
    _name = 'school.teacher'
    _description = 'Los profesores'

    name = fields.Char() # Todos los modelos deben tener un field name
    topic = fields.Char()
    phone = fields.Char()

    classrooms = fields.Many2many(comodel_name='school.classroom',
                                   relation='teachers_classroom',
                                   column1='teacher_id',
                                   column2='classroom_id')
    classrooms_last_year = fields.Many2many(comodel_name='school.classroom',
                                             relation='teachers_classroom_ly',
                                             column1='teacher_id',
                                             column2='classroom_id')

```

- La vista tree de profesor, por defecto sólo muestra el nombre. Debemos hacer la vista tree de los profesores.

```

<record model="ir.ui.view" id="school.teacher_list">
    <field name="name">school.teacher list</field>
    <field name="model">school.teacher</field>
    <field name="arch" type="xml">
        <tree>
            <field name="name"/>
            <field name="topic"/>
            <field name="phone"/>
        </tree>
    </field>
</record>

```

- De esta forma obtenemos:

school student window / Fannie Tidbold

[Editar](#) [Crear](#) [Acción](#) 1 / 1

Nombre	Fannie Tidbold	Inscription Date	17/02/2021 13:16:28
Birth Year	2.003	Last Login	
Password	G-O9lrMHFIUv3F7N	Is Student	
DNI	77516174X	Classroom	DAM2
Description			

[Teachers](#)

Name	Topic	Phone
José Alberto		
José Luis		

- Pero imaginemos que en la vista del estudiante, y sólo en la del estudiante, no queremos que aparezca el teléfono del profesor. La forma sería:

```
<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <field name="photo" widget="image"/>
                <group>
                    <group>
                        <field name="name"/>
                        <field name="birth_year"/>
                        <field name="password"/>
                        <field name="dni"/>
                        <field name="description"/>
                    </group>
                    <group>
                        <field name="inscription_date"/>
                        <field name="last_login"/>
                        <field name="is_student"/>
                        <field name="classroom"/>
                    </group>
                </group>
                <notebook>
                    <page string="Teachers">
                        <field name="teachers">
                            <tree>
                                <field name="name"/>
                                <field name="topic"/>
                            </tree>
                        </field>
                    </page>
                </notebook>
            </sheet>
        </form>
    </field>
</record>
```

- Con esto conseguimos:

school student window / Fannie Tidbold

[Editar](#) [Crear](#) [Acción ▾](#) [1 / 1](#) [◀](#)



Nombre	Fannie Tidbold	Inscription Date	17/02/2021 13:16:28
Birth Year	2.003	Last Login	
Password	G-O9!rMHFiUv3F7N	Is Student	
DNI	77516174X	Classroom	DAM2
Description			

[Teachers](#)

Name	Topic
José Alberto	
José Luis	

- Pero en la vista tree del profesor tenemos:

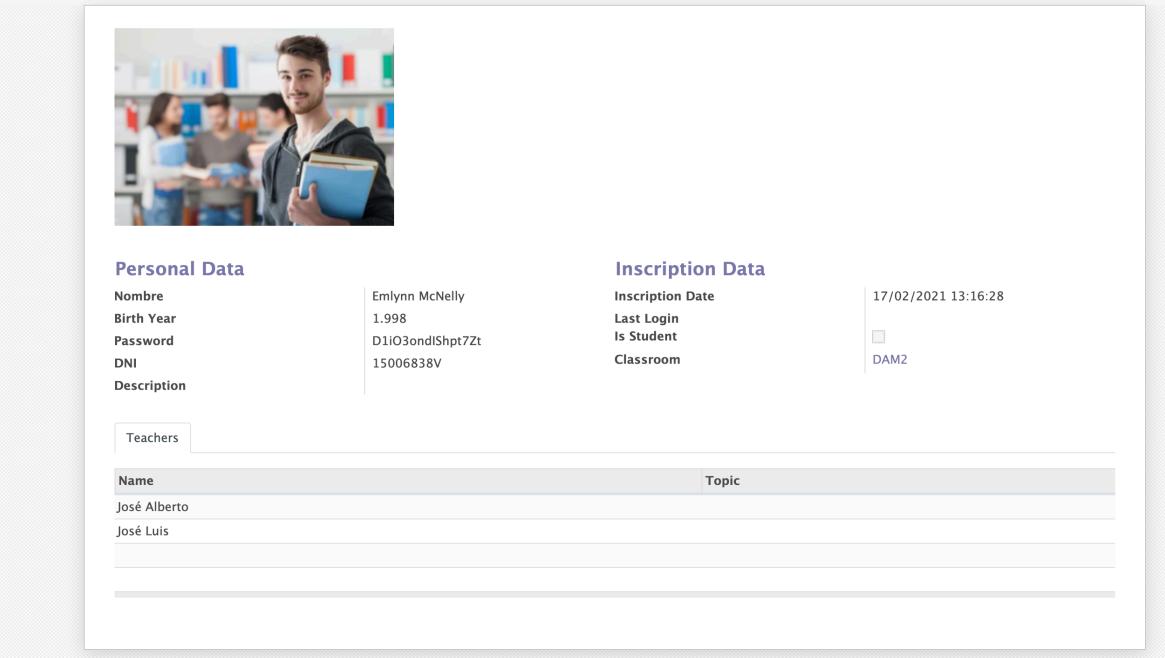
school teacher window

[Crear](#) [Importar](#) [▲](#) [Buscar...](#) [▼ Filtros](#) [≡ Agrupar por](#) [★ Favoritos](#) [1-2 / 2](#) [◀](#)

<input type="checkbox"/> Name	Topic	Phone
<input type="checkbox"/> José Alberto		
<input type="checkbox"/> José Luis		

- Vamos ahora a ver otra etiqueta que es <separator>.

```
<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <field name="photo" widget="image"/>
                <group>
                    <group>
                        <separator string="Personal Data"></separator>
                        <field name="name"/>
                        <field name="birth_year"/>
                        <field name="password"/>
                        <field name="dni"/>
                        <field name="description"/>
                    </group>
                    <group>
                        <separator string="Inscription Data"></separator>
                        <field name="inscription_date"/>
                        <field name="last_login"/>
                        <field name="is_student"/>
                        <field name="classroom"/>
                    </group>
                </group>
                <notebook>
                    <page string="Teachers">
                        <field name="teachers">
                            <tree>
                                <field name="name"/>
                                <field name="topic"/>
                            </tree>
                        </field>
                    </page>
                </notebook>
            </sheet>
        </form>
    </field>
</record>
```



The screenshot shows a student profile for 'Emlynn McNelly'. The interface includes a photo of the student, a sidebar with 'Teachers' listed, and sections for 'Personal Data' and 'Inscription Data'.

Personal Data		Inscription Data	
Nombre	Emlynn McNelly	Inscription Date	17/02/2021 13:16:28
Birth Year	1.998	Last Login	
Password	D1o3ondlShpt7Zt	Is Student	
DNI	15006838V	Classroom	DAM2
Description			

Teachers:

Name	Topic
José Alberto	
José Luis	

- Vamos a establecer el password de tipo password.

```

<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <field name="photo" widget="image"/>
                <group>
                    <group>
                        <separator string="Personal Data"></separator>
                        <field name="name"/>
                        <field name="birth_year"/>
                        <field name="password" password="True"/>
                        <field name="dni"/>
                        <field name="description"/>
                    </group>
                    <group>
                        <separator string="Inscription Data"></separator>
                        <field name="inscription_date"/>
                        <field name="last_login"/>
                        <field name="is_student"/>
                        <field name="classroom"/>
                    </group>
                </group>
            </sheet>
        </form>
    </field>
</record>

```

[Editar](#)[Crear](#)

Acción ▾

1 / 1



Personal Data

Nombre	Emlynn McNelly
Birth Year	1.998
Password	*****
DNI	15006838V
Description	

Inscription Data

Inscription Date	17/02/2021 13:16:28
Last Login	
Is Student	<input type="checkbox"/>
Classroom	DAM2

Teachers

Name	Topic
José Alberto	
José Luis	

- Vamos a modificar el modelo **classroom**. Fields.Selection es un desplegable que permite elegir una opción que es la que se guarda en la BDD. Recibe una lista de tuplas, el primer valor de cada dupla es lo que se guarda en la BDD y el segundo el texto que muestra.

```

class classroom(models.Model):
    _name = 'school.classroom'
    _description = 'Las clases'

    name = fields.Char() # Todos los modelos deben tener un field name
    level = fields.Selection([('1','1'),('2','2')])
    students = fields.One2many(string='Alumnos', comodel_name='school.student', )
    teachers = fields.Many2many(comodel_name='school.teacher',
                                 relation='teachers_classroom',
                                 column1='classroom_id',
                                 column2='teacher_id')
    teachers_last_year = fields.Many2many(comodel_name='school.teacher',
                                         relation='teachers_classroom_ly',
                                         column1='classroom_id',
                                         column2='teacher_id')
    # Vamos a considerar para el ejemplo que una clase puede tener un coordinador
    # pudiera ser coordinador de varias clases
    coordinator = fields.Many2one('school.teacher', compute='_get_coordinator')

    all_teachers = fields.Many2many('school.teacher', compute='_get_teacher')

    def _get_coordinator(self):
        for classroom in self:
            if len(classroom.teachers) > 0:
                classroom.coordinator = classroom.teachers[0].id #Para el ejemplo
            else:
                classroom.coordinator = None

```

school classroom window / Nuevo

[Guardar](#) [Descartar](#)

Name	Alumnos	Level		
Nombre	Apellido	1 2		
Agregar línea				
0				
Teachers	Name	Topic	Phone	
Agregar línea				
0				
Teachers Last Year	Name	Topic	Phone	
Agregar línea				
0				
Coordinator	All Teachers	Name	Topic	Phone
Agregar línea				
0				

- Si editamos la clase **2DAM**, le asignamos level 2, y aparece:

school classroom window / DAM2

Editar Crear Acción 1 / 1

Name Alumnos	DAM2	Level	2
Nombre	Password	Birth Year	
Vicki Cunnell	SKL2UaSXYSqj34Uj	1.967	
Emlynn McNelly	D1lO3ondlShpt7Zt	1.998	
Anni Dutt	rpwOYvQlvNmrlwZ	1.995	
Selig Ower	qTAgDjbTpqzAwYUd	2.010	
Jarrett Holbie	4936Ut8FjxZwHyyG	2.004	
Hazel Loyley	VwJjfufXcaYXZfk0	2.011	
eefsfdf	sdfsdfsdf	2.000	
Fannie Tidbold	G-091rMHFiUv3F7N	2.003	
Dud Lewcock	Wo6GIegMZJNN7e1T	2.012	
Lyon Kempton	TfaTINMWPmbl-cw	1.985	
Nickie Saphred	Jqvawbw5ETqPDGr8	1.997	
Domingo Shakespeare	eY6HJWhBMUGhevVZ	1.997	
Cayla Iacovino	YqZMsUFjfPRKb_VF	1.976	
Coraly Elwin	w8nrDcRgQ_-jdM5I	2.006	
Farris Isaacs	BW556R_MDr6xHPSh	1.998	
Lilias Jasper	Cigc0oYqmfatmr_g	1.992	

- Vamos a crear más clases DAM1, ASIR1 y ASIR2:

School Management

school classroom window

Crear Importar Filtre Buscar.

<input type="checkbox"/> Name
<input type="checkbox"/> DAM2
<input type="checkbox"/> DAM1
<input type="checkbox"/> ASIR1
<input type="checkbox"/> ASIR2

- Vamos a hacer un **level** también en el estudiante.

```

class student(models.Model):
    _name = 'school.student'
    _description = 'Los alumnos'

    name = fields.Char(string="Nombre", readonly=False, required=True, help='Este es el nombre')
    birth_year = fields.Integer()
    dni = fields.Char(string="DNI")
    password = fields.Char(default=lambda p: secrets.token_urlsafe(12))

    description = fields.Text()
    inscription_date = fields.Datetime(default=lambda d: fields.Datetime.now())
    last_login = fields.Datetime()
    is_student = fields.Boolean()
    level = fields.Selection([('1', '1'), ('2', '2')])
    photo = fields.Image(max_width=300, max_height=300) # Field binario pero específico para imágenes.
    classroom = fields.Many2one('school.classroom', ondelete='set null', help='Clase a la que pertenece')
    teachers = fields.Many2many('school.teacher', related='classroom.teachers', readonly=True)

    @api.constraints('dni')
    def _check_dni(self):
        regex = re.compile('^\d{9}|\d{8}[a-zA-Z]\d$', re.I)
        for student in self:
            #Ahora vamos a validar si se cumple la condición
            if not regex.match(student.dni):
                #No coinciden por lo que tenemos que informar e impedir que se guarde
                raise ValidationError('DNI format incorrect')

    _sql_constraints = [('dni_uniq', 'unique(dni)', 'DNI can\'t be repeated')]

```

- También lo tenemos que añadir en la vista del estudiante.

```

<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <field name="photo" widget="image"/>
                <group>
                    <group>
                        <separator string="Personal Data"></separator>
                        <field name="name"/>
                        <field name="birth_year"/>
                        <field name="password" password="True"/>
                        <field name="dni"/>
                        <field name="description"/>
                    </group>
                    <group>
                        <separator string="Inscription Data"></separator>
                        <field name="inscription_date"/>
                        <field name="last_login"/>
                        <field name="is_student"/>
                        <field name="classroom"/>
                        <field name="level"/>
                    </group>
                </group>
            </sheet>
        </form>
    </field>
</record>

```

- Si refrescamos, tal y como lo tenemos ahora, nada nos impide seleccionar 1ASIR en un alumno e indicar que es de nivel 2 y esto no debería permitirse.

school student window / Hazel Loyley

Editar Crear Acción ▾

Personal Data		Inscription Data	
Nombre	Hazel Loyley	Inscription Date	17/02/2021 13:16:28
Birth Year	2.011	Last Login	
Password	*****	Is Student	True
DNI	05492519S	Classroom	ASIR1
Description		Level	2

Teachers

Name Topic

- Para solucionar esto, vamos a aplicar un filtro en estudiante en el **Many2one** de **classroom**. **Domain** se utiliza para muchas cosas, ya que es lo que utiliza **Odoo** para hacer consultas **sql**. Recibe una lista de tuplas, cada tupla tiene tres elementos, el primer campo se refiere al level remoto (de la clase relacionada) y el segundo level al actual. El actual (no remoto) va sin comilla.

```
class student(models.Model):
    _name = 'school.student'
    _description = 'Los alumnos'

    name = fields.Char(string="Nombre", readonly=False, required=True, help='Este es el nombre')
    birth_year = fields.Integer()
    dni = fields.Char(string="DNI")
    password = fields.Char(default=lambda p: secrets.token_urlsafe(12))

    description = fields.Text()
    inscription_date = fields.Datetime(default=lambda d: fields.Datetime.now())
    last_login = fields.Datetime()
    is_student = fields.Boolean()
    level = fields.Selection([('1','1'),('2','2')])
    photo = fields.Image(max_width=300, max_height=300, # Field binario pero específico para imágenes.
    classroom = fields.Many2one('school.classroom', domain="[(level,'=',level)]", ondelete='set null', help='Clase a la que pertenece')
    teachers = fields.Many2many('school.teacher', related='classroom.teachers', readonly=True)

    @api.constrains('dni')
    def _check_dni(self):
        regex = re.compile('^\d{8}[a-z]\d$', re.I)
        for student in self:
            if not regex.match(student.dni):
                raise ValidationError(_('El DNI no es válido'))
```

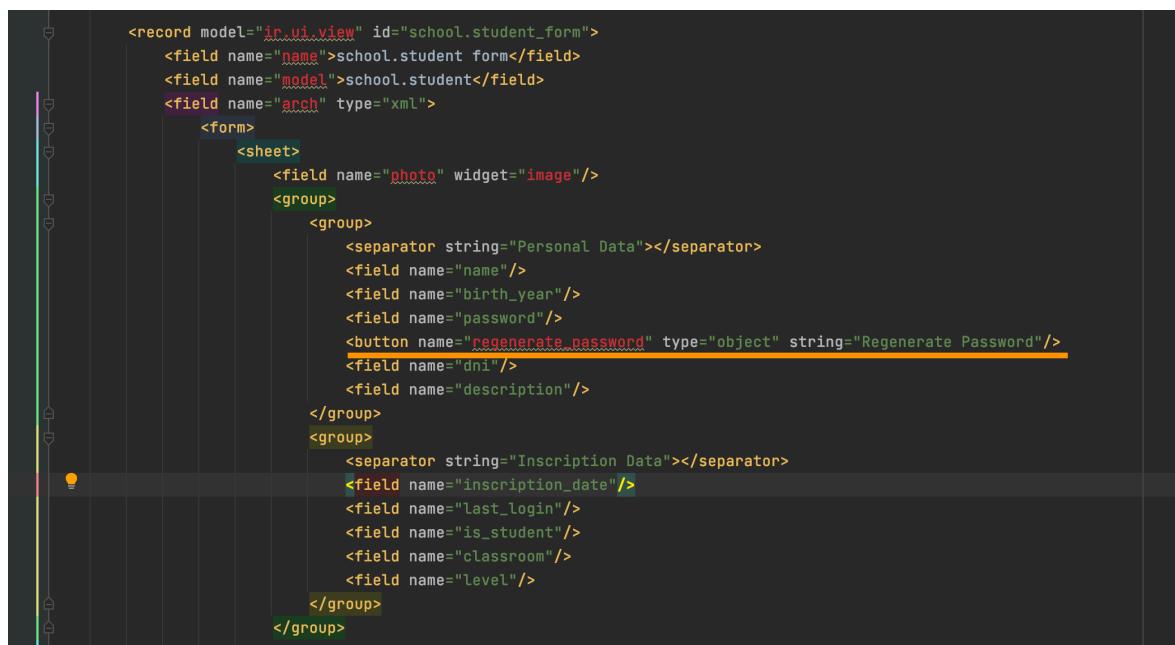
- Este código puede estar en el modelo, o también se puede añadir como valor en la vista:

```

<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <field name="photo" widget="image"/>
                <group>
                    <group>
                        <separator string="Personal Data"></separator>
                        <field name="name"/>
                        <field name="birth_year"/>
                        <field name="password" password="True"/>
                        <field name="dni"/>
                        <field name="description"/>
                    </group>
                    <group>
                        <separator string="Inscription Data"></separator>
                        <field name="inscription_date"/>
                        <field name="last_login"/>
                        <field name="is_student"/>
                        <field name="classroom" domain=" [('level', '=', level)] "/>
                        <field name="level"/>
                    </group>
                </group>
            </sheet>
        </form>
        <notebook>
            <page string="Teachers">

```

- Si lo ponemos en el modelo, se aplicará directamente en el modelo y no sólo cuando se muestre en la vista. Es decisión del desarrollador según el caso.
- Hay muchos más widgets que se pueden utilizar, podemos mirar la documentación para ver las opciones que tenemos.
- Ahora vamos a añadir botones para añadir funcionalidad, automatizaciones, etc.
- Vamos a añadir un botón que permita regenerar la contraseña de un estudiante.
- Vamos a eliminar el tipo de campo **password** para poder ver que el cambio se produce correctamente.



```

<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <field name="photo" widget="image"/>
                <group>
                    <group>
                        <separator string="Personal Data"></separator>
                        <field name="name"/>
                        <field name="birth_year"/>
                        <field name="password"/>
                        <button name="regenerate_password" type="object" string="Regenerate Password"/>
                        <field name="dni"/>
                        <field name="description"/>
                    </group>
                    <group>
                        <separator string="Inscription Data"></separator>
                        <field name="inscription_date"/>
                        <field name="last_login"/>
                        <field name="is_student"/>
                        <field name="classroom"/>
                        <field name="level"/>
                    </group>
                </group>
            </sheet>
        </form>
        <notebook>

```

- Este nombre se tiene que corresponder con una función en el modelo que le permita gestionar el comportamiento cuando es pulsado.

school student window / Selig Ower

[Editar](#) [Crear](#) [Acción ▾](#)



Personal Data		Inscription Data	
Nombre	Selig Ower	Inscription Date	17/02/2021 13:16:28
Birth Year	2.010	Last Login	
Password	qTAgDjbTpqzAwYUd	Is Student	
Regenerate Password		Classroom	DAM2
DNI	82474061Z	Level	2
Description			

[Teachers](#)

Name	Topic
José Alberto	
José Luis	

- Queremos que cada vez que pulsemos el botón se regenere la contraseña y se guarde en la BDD automáticamente:

```

class student(models.Model):
    _name = 'school.student'
    _description = 'Los alumnos'

    name = fields.Char(string="Nombre", readonly=False, required=True, help='Este es el nombre')
    birth_year = fields.Integer()
    dni = fields.Char(string="DNI")
    password = fields.Char(default=lambda p: secrets.token_urlsafe(12))

    description = fields.Text()
    inscription_date = fields.Datetime(default=lambda d: fields.Datetime.now())
    last_login = fields.Datetime()
    is_student = fields.Boolean()
    level = fields.Selection([('1', '1'), ('2', '2')])
    photo = fields.Image(max_width=300, max_height=300) # Field binario pero específico para imágenes
    classroom = fields.Many2one('school.classroom', domain="['Level', '=', level]", ondelete='set null')
    teachers = fields.Many2many('school.teacher', related='classroom.teachers', readonly=True)

    @api.constrains('dni')
    def _check_dni(self):
        regex = re.compile('[0-9]{8}[a-zA-Z]', re.I)
        for student in self:
            #Ahora vamos a validar si se cumple la condición
            if not regex.match(student.dni):
                #No coinciden por lo que tenemos que informar e impedir que se guarde
                raise ValidationError('DNI format incorrect')

    _sql_constraints = [('dni_uniq','unique(dni)','DNI can\'t be repeated')]

    # También recibe un recordset
    def regenerate_password(self):
        for student in self:
            pw = secrets.token_urlsafe(12)
            student.write({'password': pw})

```

- Si reiniciamos el servicio, comprobaremos que al pulsar el botón regenera una nueva contraseña y queda refrescada en el formulario.
- Para que quede mejor el botón, lo vamos a añadir dentro de un reader dentro de sheet. De esta forma queda dentro del formulario, si lo ponemos fuera del sheet queda fuera del formulario, como podemos ver.

```
<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <header>
                <button name="regenerate_password" type="object" string="Regenerate Password"/>
            </header>
            <sheet>
                <field name="photo" widget="image"/>
                <group>
                    <group>
                        <separator string="Personal Data"></separator>
                        <field name="name"/>
                        <field name="birth_year"/>
                        <field name="password"/>
                        <field name="dni"/>
                        <field name="description"/>
                    </group>
                    <group>
                        <separator string="Inscription Data"></separator>
                        <field name="inscription_date"/>
                        <field name="last_login"/>
                        <field name="is_student"/>
                        <field name="classroom"/>
                        <field name="level"/>
                    </group>
                </group>
                <notebook>
                    <page string="Teachers">
                        <field name="teachers">
                            <tree>
                                <field name="name"/>
                                <field name="topic"/>
                            </tree>
                        </field>
                    </page>
                </notebook>
            </sheet>
        </form>
    </field>
</record>
```

school student window / Selig Ower

Editar Crear Acción ▾ 1 / 1 < >

Regenerate Password



Personal Data

Nombre	Selig Ower
Birth Year	2.010
Password	LCQDcg5zq3hHjM-T
DNI	82474061Z
Description	

Inscription Data

Inscription Date	17/02/2021 13:16:28
Last Login	
Is Student	
Classroom	DAM2
Level	2

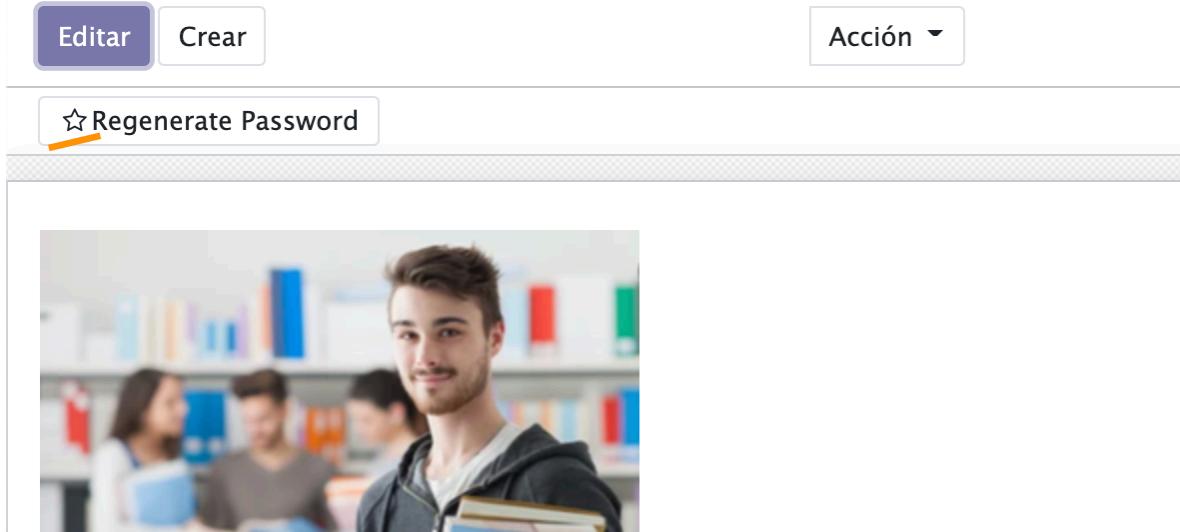
Teachers

Name	Topic
José Alberto	
José Luis	

- En esta URL: <https://es.slideshare.net/TaiebKristou/odoo-icon-smart-buttons>
Podéis encontrar iconos propios de ODOO.

```
views.xml x models.py x Add Configuration... 33 32 ✘
26
27
28
29 <record model="ir.ui.view" id="school.student_form">
30   <field name="name">school.student form</field>
31   <field name="model">school.student</field>
32   <field name="arch" type="xml">
33     <form>
34       <header>
35         <button name="regenerate_password" type="object" icon="fa-star-o" string="Regenerate Password"/>
36       </header>
37       <sheet>
38         <field name="photo" widget="image"/>
39         <group>
40           <group>
41             <separator string="Personal Data"></separator>
42             <field name="name"/>
43             <field name="birth_year"/>
44             <field name="password"/>
45             <field name="dni"/>
46             <field name="description"/>
47           </group>
48           <group>
49             <separator string="Inscription Data"></separator>
50             <field name="inscription_date"/>
51             <field name="last_login"/>
52           </group>
53         </group>
54       </sheet>
55     </form>
56   </field>
57 </record>
```

school student window / Selig Ower



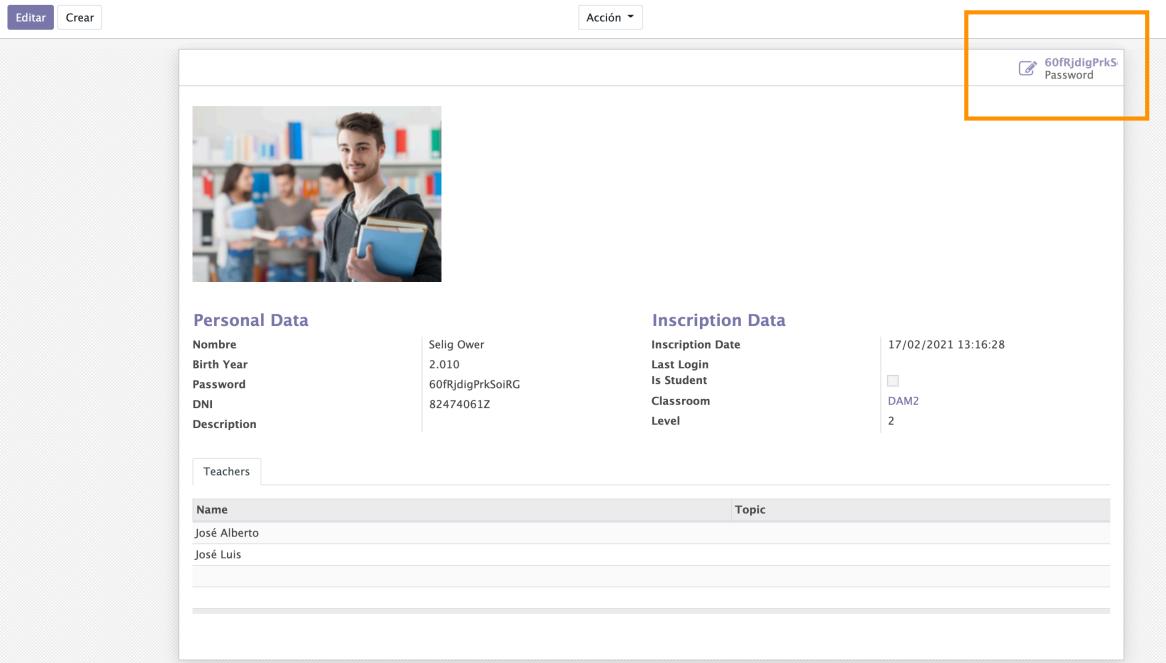
- Vamos a ver ahora qué es un **smart button**:

```
<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <div class="oe_button_box">
                    <button type="object" class="oe_stat_button" icon="fa-pencil-square-o"
                           name="regenerate_password">
                        <div class="o_form_field o_stat_info">
                            <span class="o_stat_value">
                                <field name="password" string="Password"/>
                            </span>
                            <span class="o_stat_text">Password</span>
                        </div>
                    </button>
                </div>
                <field name="photo" widget="image"/>
            <group>
                <group>
                    <separator string="Personal Data"></separator>
                    <field name="name"/>
                    <field name="birth_year"/>
                    <field name="password"/>
                    <field name="dni"/>
                    <field name="description"/>
                </group>
                <group>
                    <separator string="Inscription Data"></separator>
                    <field name="inscription_date"/>
                    <field name="last_inscription"/>
                </group>
            </group>
        </form>
    </field>
</record>
```

The XML code defines a view for "school.student_form". It includes fields for the name and model, and a large block of XML arch. The arch contains a form with a sheet section. Inside the sheet, there is a div with a class of "oe_button_box". Within this box, there is a button of type "object" with a class of "oe_stat_button" and an icon of "fa-pencil-square-o". The button is named "regenerate_password". Inside the button's click handler, there is a div with a class of "o_form_field o_stat_info", containing a span with a class of "o_stat_value" (containing a password field) and a span with a class of "o_stat_text" (containing the text "Password"). The entire "oe_button_box" div is highlighted with an orange box. The code continues with other fields like "photo" and groups of fields for personal and inscription data.

- Esta estructura de divs, button, span, etc con estas clases da como resultado lo siguiente:

Editar Crear Acción ▾



Personal Data		Inscription Data	
Nombre	Selig Ower	Inscription Date	17/02/2021 13:16:28
Birth Year	2.010	Last Login	
Password	60fRjdigPrkSoiRG	Is Student	DAM2
DNI	82474061Z	Classroom	
Description		Level	2

Teachers

Name	Topic
José Alberto	
José Luis	

- Esto sería lo más recomendable a la hora de crear botones para añadir funcionalidad a un formulario. Es la forma más elegante en las versiones actuales de **Odoo**.
- En este ejemplo muestra sólo una parte del password porque tiene unas dimensiones concretas.
- Ahora vamos a aprender a **ocultar campos condicionalmente**. Vamos a ocultar el campo classroom en el caso en el que no esté activo el campo **Is Student**.

```
views.xml x models.py x
  39
  40
  41
  42
  43
  44
  45
  46
  47
  48
  49
  50
  51
  52
  53
  54
  55
  56
  57
  58
  59
  60
  61
  62
  63
  64
  65
  66
  67
  68
  69
  70
  71
  72
```

```
<span class="o_stat_value">
    <field name="password" string="Password"/>
</span>
<span class="o_stat_text">Password</span>
</div>
</button>
</div>
<field name="photo" widget="image"/>
<group>
    <group>
        <separator string="Personal Data"></separator>
        <field name="name"/>
        <field name="birth_year"/>
        <field name="password"/>
        <field name="dni"/>
        <field name="description"/>
    </group>
    <group>
        <separator string="Inscription Data"></separator>
        <field name="inscription_date"/>
        <field name="last_login"/>
        <field name="is_student"/>
        <field name="classroom"
            attrs="{'invisible':[( 'is_student', '=', False)]}"/>
        <field name="level"/>
    </group>
</group>
<notebook>
    <page string="Teachers">
        <field name="teachers">
            <tree>
                <field name="name"/>
                <field name="topic"/>
            </tree>
        </field>
    </page>
</notebook>
```

 D1iO3ondISI
Password



Personal Data

Nombre	Emlynn McNelly
Birth Year	1.998
Password	D1iO3ondIShpt7Zt
DNI	15006838V
Description	

Inscription Data

Inscription Date	17/02/2021 13:16:28
Last Login	
Is Student	<input type="checkbox"/>
Level	

Teachers

Name	Topic
José Alberto	
José Luis	

 Password



Personal Data

Nombre	Emlynn McNelly
Birth Year	1.998
Password	D1iO3ondIShpt7Zt
DNI	15006838V
Description	

Inscription Data

Inscription Date	17/02/2021 13:16:28
Last Login	
Is Student	<input checked="" type="checkbox"/>
Classroom	DAM2
Level	

Teachers

Name	Topic
José Alberto	
José Luis	

- Hay otra cosa interesante referente al **header**. El **header** acepta un field que se debe llamar **state**. Además debe tener un **widget statusbar**. Como el campo **status** no lo tenemos, debemos añadirlo en el modelo.

The screenshot shows a code editor with two tabs: 'views.xml' and 'models.py'. The 'views.xml' tab displays an XML configuration for a 'student' form. The XML includes fields for 'topic' and 'phone', and a detailed view for 'school.student_form' with a header containing a 'state' field. The 'models.py' tab is partially visible at the top.

```
<field name="topic"/>
<field name="phone"/gt;
</tree>
</field>
</record>

<record model="ir.ui.view" id="school.student_form">
    <field name="name">school.student form</field>
    <field name="model">school.student</field>
    <field name="arch" type="xml">
        <form>
            <header>
                <field name="state" widget="statusbar"></field>
            </header>
            <sheets>
                <div class="oe_button_box">
                    <button type="object" class="oe_stat_button" icon="fa-pencil-square-o">
                        <name="regenerate_password">
                            <div class="o_form_field o_stat_info">
                                <span class="o_stat_value">
                                    <field name="password" string="Password"/>
                                </span>
                                <span class="o_stat_text">Password</span>
                            </div>
                        </button>
                    </div>
                    <field name="photo" widget="image"/gt;
                <group>
                    <group>
```

The screenshot shows a code editor with two tabs: 'views.xml' and 'models.py'. The 'models.py' tab is active, displaying Python code for a 'student' model. The code includes fields for name, birth year, DNI, password, description, inscription date, last login, is_student, level, photo, classroom, teachers, state, and a constructor. It also contains a check_dni method that uses a regular expression to validate the DNI number.

```
views.xml x models.py x
9 _logger = logging.getLogger(__name__)
10
11
12 class student(models.Model):
13     _name = 'school.student'
14     _description = 'Los alumnos'
15
16     name = fields.Char(string="Nombre", readonly=False, required=True, help='Este es el nombre')
17     birth_year = fields.Integer()
18     dni = fields.Char(string="DNI")
19     password = fields.Char(default=lambda p: secrets.token_urlsafe(12))
20
21     description = fields.Text()
22     inscription_date = fields.Datetime(default=lambda d: fields.Datetime.now())
23     last_login = fields.Datetime()
24     is_student = fields.Boolean()
25     level = fields.Selection([('1', '1'), ('2', '2')])
26     photo = fields.Image(max_width=300, max_height=300) # Field binario pero específico para imágenes.
27     classroom = fields.ManyZone('school.classroom', domain="[(level, '=', level)]", ondelete='set null', help='Clase a la que')
28     teachers = fields.Many2many('school.teacher', related='classroom.teachers', readonly=True)
29     state = fields.Selection([('1', 'Matriculado'), ('2', 'Estudiante'), ('3', 'Ex-estudiante')], default="1")
30     def __init__(self):
31         super().__init__()
32     def _check_dni(self):
33         regex = re.compile(r'^[0-9]{8}[a-zA-Z]$', re.I)
34         for student in self:
35             #Ahora vamos a validar si se cumple la condición
36             if not regex.match(student.dni):
37                 #No coinciden por lo que tenemos que informar e impedir que se guarde
38                 raise ValidationError('DNI format incorrect')
```

- De este modo obtenemos lo siguiente (he tenido que crear un nuevo estudiante, y al editarlo aparece arriba):

Editar Crear Acción ▾ 2 / 2 < >

Matriculado Estudiante Ex-estudiante

w2K9ZbuYnv
password

Personal Data

Nombre	Luis
Birth Year	0
Password	w2K9ZbuYnvuFASj
DNI	12345678B
Description	

Inscription Data

Inscription Date	23/02/2021 12:05:13
Last Login	
Is Student	<input checked="" type="checkbox"/>
Classroom	DAM1
Level	1

Teachers

Name	Topic

- Por defecto aparece en esa situación y sería como un asistente, podemos añadir un botón para que pase de un estado a otro.
- También podemos hacer que un grupo de campos se muestren o no en función del estado en el que está. Por ejemplo:

The screenshot shows a code editor with two tabs: 'views.xml' and 'models.py'. The 'views.xml' tab contains XML code for a user interface. The code includes sections for a header, a sheet, and a notebook. It features various fields like 'name', 'birth_year', 'password', 'dni', 'description', 'inscription_date', 'last_login', 'is_student', and 'level'. A specific group of fields is highlighted with a yellow background, indicating it is only visible in state 2. The 'models.py' tab is partially visible at the top.

```

<header>
    <field name="state" widget="statusbar"></field>
</header>
<sheet>
    <div class="oe_button_box">
        <button type="object" class="oe_stat_button" icon="fa-pencil-square-o"
            name="regenerate_password">
            <div class="o_form_field o_stat_info">
                <span class="o_stat_value">
                    <field name="password" string="Password"/>
                </span>
                <span class="o_stat_text">Password</span>
            </div>
        </button>
    </div>
    <field name="photo" widget="image"/>
    <group>
        <group>
            <separator string="Personal Data"></separator>
            <field name="name"/>
            <field name="birth_year"/>
            <field name="password"/>
            <field name="dni"/>
            <field name="description"/>
        </group>
        <group states="2">
            <separator string="Inscription Data"></separator>
            <field name="inscription_date"/>
            <field name="last_login"/>
            <field name="is_student"/>
            <field name="classroom"
                attrs="{'invisible':[('is_student','=',False)]}"/>
            <field name="level"/>
        </group>
    </group>
    <notebook>

```

- El grupo sólo se mostrará cuando esté en el estado 2. Como está en el 1, no se muestra.

school student window / Luis

The screenshot shows the Odoo application interface for a student named 'Luis'. At the top, there are buttons for 'Editar' (Edit) and 'Crear' (Create), a dropdown menu for 'Acción', and a status bar showing '1 / 1' and navigation arrows. Below the header, there is a photo of Luis holding a blue book. The main area is divided into sections: 'Personal Data' (Nombre: Luis, Birth Year: 0, Password: w2K9ZbuYnwsuFASj, DNI: 123456788) and a 'Teachers' section (empty). At the bottom, there is a 'Topic' section with a table and several empty rows.

Name	Topic

- Lo quitamos para que se muestre, porque sólo era un ejemplo.

- Podemos añadir más atributos condicionales, por ejemplo:

```
<group>
    <separator string="Personal Data"></separator>
    <field name="name"/>
    <field name="birth_year"/>
    <field name="password"/>
    <field name="dni"/>
    <field name="description"/>
</group>
<group>
    <separator string="Inscription Data"></separator>
    <field name="inscription_date"/>
    <field name="last_login"/>
    <field name="is_student"/>
    <field name="classroom"
        attrs="{'invisible':[(('is_student','='),False)],
                'required':[(('is_student','='),True)]}"/>
    <field name="level"/>
</group>
</group>
<notebook>
```

The screenshot shows a user interface for managing personal and inscription data. At the top, there is a placeholder image of a student holding books. Below it, the 'Personal Data' section contains fields for Name (Lyon Kempton), Birth Year (1.985), Password (TfaTINMWPmbbl-cw), DNI (84444777R), and Description (empty). The 'Inscription Data' section includes fields for Inscription Date (17/02/2021 13:16:29), Last Login (empty), Is Student (checked), Classroom (DAM2), and Level (empty). A red circle highlights the 'Level' dropdown menu, which contains the value 'DAM2'. At the bottom left, there is a 'Teachers' tab and a table showing names (José Alberto, José Luis) and topics (empty).

- También podríamos añadir la propiedad **readonly**.