

Crear una funcion para el controlador de odoo

1.- Crear una funcion para calcular la edad ,de los alumnos en prácticas Nota 5

2.- Para poder llegar al 10 , crear otras 2 funciones y realizar un manual explicando como se han realizado y su objetivo.

```
from odoo import models, fields, api
from datetime import date

class AlumnoPracticas(models.Model):
    _name = 'alumno.practicas'
    _description = 'Alumnos en Prácticas'

    name = fields.Char(string='Nombre', required=True)
    birth_date = fields.Date(string='Fecha de Nacimiento', required=True)
    edad = fields.Integer(string='Edad', compute='_compute_edad', store=True)

    @api.depends('birth_date')
    def _compute_edad(self):
        for record in self:
            if record.birth_date:
                today = date.today()
                record.edad = today.year - record.birth_date.year - ((today.month, today.day) < (record.birth_date.month, record.birth_date.day))
            else:
                record.edad = 0

    def get_students_above_age(self, age):
        """
        Obtiene todos los alumnos cuya edad sea mayor a la edad dada.
        """
        return self.search([('edad', '>', age)])

    def format_student_data(self):
        """
        Retorna un diccionario con los datos formateados de los alumnos.
        """
        students_data = {}
        for student in self.search([]):
            students_data[student.name] = {
                'Edad': student.edad,
                'Fecha de Nacimiento': student.birth_date
            }
        return students_data
```

Explicación del Código

Este código pertenece a un módulo de Odoo que gestiona la información de los alumnos en prácticas. He creado tres funciones en el modelo alumno.practicas, cada una con un propósito específico.

1. _compute_edad(self)

Objetivo: Calcular la edad de los alumnos en función de su fecha de nacimiento.

Explicación:

- Odoo tiene un campo birth_date donde se almacena la fecha de nacimiento del alumno. La función toma la fecha actual (date.today()) y la compara con la fecha de nacimiento.
- Usa una operación matemática para ajustar la edad si el cumpleaños del alumno aún no ha ocurrido en el año actual.
- Guarda el resultado en el campo edad, que se actualiza automáticamente.

2. `get_students_above_age(self, age)`

Objetivo: Obtener una lista de alumnos cuya edad sea mayor a un valor dado.

Explicación:

- La función usa `self.search([('edad', '>', age)])` para encontrar alumnos con edad superior al valor pasado como parámetro.
- Devuelve una lista de registros que cumplen con esa condición.
- Es útil para filtrar alumnos mayores de cierta edad en el sistema.

3. `format_student_data(self)`

Objetivo: Formatear y devolver los datos de los alumnos en un diccionario.

Explicación:

- La función recorre todos los alumnos almacenados en la base de datos.
- Para cada alumno, guarda su nombre como clave en un diccionario y su edad y fecha de nacimiento como valores.
- Este formato hace que los datos sean fáciles de leer y usar en informes o en la interfaz de Odoo.