



PROYECTO SGE

CFGS Desarrollo de Aplicaciones Multiplataforma
Informática y Comunicaciones

Herencia en Odoo

Fecha de presentación: 4/4/2025

Nombre y Apellidos: Pablo Villagrán

Email: pablo.vilgon@educa.jcyl.es

Índice

Introducción	3
Estado del arte.....	3
Descripción general del proyecto.....	4
Objetivos.....	4
Entorno de trabajo (tecnologías de desarrollo y herramientas)	4
Documentación técnica: análisis, diseño, implementación, pruebas y despliegue	5
Análisis del sistema (funcionalidades básicas de la aplicación)	5
Diseño de la base de datos	6
Implementación	6
Herencia	13
Pruebas.....	13
Manual.....	16
Manual de usuario:	16
Conclusiones y posibles ampliaciones	17
Bibliografía	19

INTRODUCCIÓN

Resumen del proyecto: explicar brevemente qué va a realizar la aplicación

Nuestra aplicación va a consistir en un módulo llamado “clinicapablo”, diseñado para gestionar una clínica de fisioterapia. De esta manera nos permitirá controlar de forma estructurada los siguientes elementos:

- Datos del paciente
- Dolencias del cliente
- Tratamientos aplicados según la dolencia
- Sesiones de cada tratamiento
- Fisioterapeutas de la clínica
- Técnicas utilizadas

ESTADO DEL ARTE

Definición de herencia en Odoo

Es un mecanismo que permite modificar o extender comportamientos de modelos, funciones y vistas existentes, facilitando así la personalización de los módulos sin necesidad de reescribir todo el código desde cero.

Tipos de herencia

- **Herencia clásica (`inherit`)**: permite extender un modelo existente añadiendo nuevos campos o funcionalidades. El más común de herencia en Odoo.
- **Herencia por delegación (`inherits`)**: establece una relación entre dos modelos, donde uno delega una parte de su comportamiento y campos al otro, manteniendo estructuras separadas pero vinculadas a la vez.
- **Herencia de vista (`inherit id`)**: se utiliza para modificar o ampliar vistas existentes sin sobrescribirlas completamente. Esto permite añadir, quitar o reorganizar elementos en la interfaz de usuario.

DESCRIPCIÓN GENERAL DEL PROYECTO

Objetivos

Descripción de lo que se ha pretendido alcanzar con el proyecto

El objetivo principal de este proyecto ha sido desarrollar un módulo personalizado en la plataforma de Odoo. Con este desarrollo pretendemos:

- Diseñar e implementar un sistema que permita registrar información detallada sobre los pacientes, dolencias y tratamientos recibidos.
- Establecer una estructura clara entre los distintos elementos de la clínica.
- Facilitar la gestión de la clínica mediante interfaces intuitivas y personalizadas, como la vista de árbol, formulario y el calendario.
- Automatizar ciertos procesos.

Entorno de trabajo (tecnologías de desarrollo y herramientas)

Explicar todas las herramientas, tecnologías, lenguajes de programación... utilizados para desarrollar el proyecto: Docker, Visual Studio, python, gestor de base de datos...

Para el desarrollo del proyecto, he utilizado diversas tecnologías y herramientas que permiten una implementación eficiente y organizada del módulo en Odoo.

- **Odoo**: framework principal sobre el que desarrollamos el módulo.
- **Python**: lenguaje de programación para la lógica del backend. Definimos los modelos, relaciones campos computados...
- **Xml**: utilizado para las vistas, menús y estructuras visuales.
- **PostgreSQL**: gestor de base de datos que viene por defecto en Odoo. Almacena toda la información relacionada.
- **Docker**: herramienta que nos facilita la instalación y ejecución de Odoo y PostgreSQL.
- **Visual Studio**: editor de código utilizado para desarrollar el módulo.
- **Navegador web**: herramienta para acceder a la interfaz de Odoo y comprobar el funcionamiento del módulo desarrollado.

DOCUMENTACIÓN TÉCNICA: ANÁLISIS, DISEÑO, IMPLEMENTACIÓN, PRUEBAS Y DESPLIEGUE

Análisis del sistema (funcionalidades básicas de la aplicación)

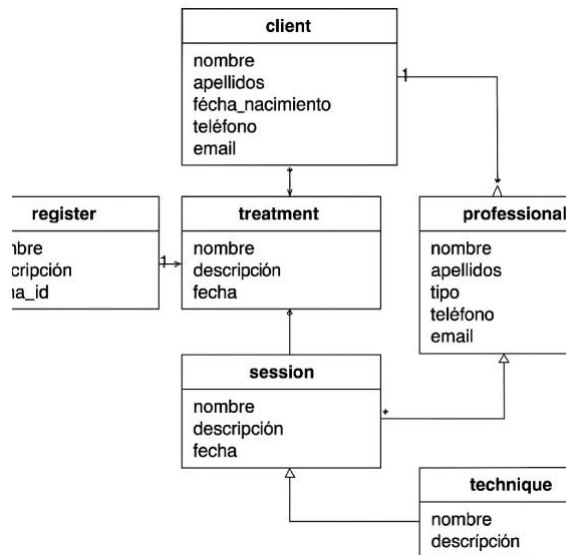
Explicar con detalle las operaciones que realiza la aplicación.

La app desarrollada nos permite gestionar de manera completa y estructurada las operaciones diarias de una clínica de fisioterapia. Sus funciones principales son:

- **Gestión de clientes**: Alta, visualización y modificación de datos personales de cada paciente, como nombre, apellido, fecha de nacimiento, teléfono, email y dirección.
- **Registro de dolencias**: asociar uno o varios registros médicos a cada cliente incluyendo nombre y descripción del problema.
- **Tratamientos**: por cada registro podemos definir múltiples tratamientos, que influyen cada nombre (combinado cliente y dolencia), descripción, fecha de inicio y fin. Cada tratamiento está a cargo de un profesional.
- **Sesiones**: cada tratamiento puede tener varias sesiones con sus respectivas fechas, descripción y técnicas empleadas. Estas se muestran en vistas tipo formulario, árbol y calendario.
- **Técnicas**: se pueden vincular estas a sesiones específicas.
- **Profesionales**: cada fisioterapeuta de la clínica con su nombre, apellidos, tipo de especialidad, email y teléfono. Incluye un campo computado que muestra todas las sesiones en las que ha intervenido
- **Vista calendario de sesiones**: donde podemos visualizar las sesiones programadas, agrupadas por fecha y coloreadas según el profesional responsable

Diseño de la base de datos

Explicar el diseño de la base de datos. Incluir un diagrama con las tablas y sus relaciones



Implementación

Incluir los siguientes apartados:

1. Captura del proceso usado para crear la estructura básica del módulo de Odoo

odoo

<

249102ffaa5d

odoo:16

8069:8069

STATUS

Running (E

Logs

Inspect

Bind mounts

Exec

Files

Stats

```

$ ls
bin boot dev entrypoint.sh etc home lib media mnt opt proc root run sbin srv
$ cd mst-addons
/bin/sh: 2: cd: can't cd to mst-addons
$ cd /mnt/extra-addons
$ ls
managepablo padelschoolpablo repaso repaso.zip
$ odoo scaffold clinicapablo
$ ls
clinicapablo managepablo padelschoolpablo repaso repaso.zip
$ cd clinicapablo
$ ls
__init__.py __manifest__.py controllers demo models security views
$

```

2. Captura de los modelos creados con los campos, relaciones y atributos correspondientes

Client:

```
client.py 1 X
models > client.py > ...
1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields
4
5  class Client(models.Model):
6      _name = 'clinicapablo.client'
7      _description = 'Client'
8
9      name = fields.Char(string="Nombre", required=True, help="Introduzca el nombre")
10     last_name = fields.Char(string="Apellidos", required=True, help="Introduzca los apellidos")
11     birth_date = fields.Date(string="Fecha nacimiento")
12     phone = fields.Char(string="Número de teléfono")
13     email = fields.Char(string="Email")
14     address = fields.Text(string="Dirección")
15
16     # Relacion One2many con el modelo 'clinicapablo.register'
17     register_id = fields.One2many(comodel_name="clinicapablo.register", inverse_name="client_id", string="Registro")
18
19     # Metodo para personalizar el nombre en listas y relaciones
20
21     def name_get(self):
22         result = []
23         for record in self:
24             name = f"Clinic{record.name or ''}"
25             result.append((record.id, name))
26         return result
27
```

Professional:

```
from odoo import models, fields, api

class Professional(models.Model):
    _name = 'clinicapablo.professional'
    _description = 'clinicapablo.professional'

    name = fields.Char(string="Nombre", required=True, help="Introduzca el nombre")
    last_name = fields.Char(string="Apellidos", required=True, help="Introduzca los apellidos")

    # Tipo de especializacion del profesional
    type = fields.Selection([
        ('muscular', 'Muscular'),
        ('deportivo', 'Deportivo'),
        ('geriatrico', 'Geriatrico'),
        ('pediatrico', 'Pediatrico')
    ], string='Tipo', required=True)

    phone = fields.Char(string="Teléfono")
    email = fields.Char(string="Email")

    # Relacion One2many con el modelo 'clinicapablo.treatment'
    treatment_id = fields.One2many('clinicapablo.treatment', 'professional_id', string='Tratamientos Asignados')

    session_summary = fields.One2many(
        'clinicapablo.session', 'professional_id',
        string='Sesiones Asignadas',
        compute='_compute_sessions'
    )

    # Metodo para calcular las sesiones asignadas al profesional
    @api.depends('treatment_id')
    def _compute_sessions(self):
        for record in self:
            record.session_summary = self.env['clinicapablo.session'].search([('professional_id', '=', record.id)])

    # Metodo para personalizar el nombre en listas y relaciones
    def name_get(self):
        result = []
        for record in self:
            name = f"Clinic {record.name or ''}"
            result.append((record.id, name))
        return result
```

Register:

```
register.py 1 X
models > register.py > ...
1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields
4  from datetime import datetime
5
6  class Register(models.Model):
7      _name = 'clinicapablo.register'
8      _description = 'clinicapablo.register'
9
10     name = fields.Char(string="Nombre", required=True, help="Introduzca el nombre")
11     address = fields.Text(string="Descripcion")
12
13     # Relacion Many2one con el modelo 'clinicapablo.client'
14     client_id = fields.Many2one("clinicapablo.client", string="Cliente", required=True, ondelete="cascade")
15
16     # Relacion One2many con el modelo 'clinicapablo.register'
17     treatment_id = fields.One2many(string="Tratamiento", comodel_name="clinicapablo.treatment", inverse_name="register_id")
18
19     # Metodo para personalizar el nombre en listas y relaciones
20     def name_get(self):
21         result = []
22         for record in self:
23             name = f"Clinic {record.name or ''}"
24             result.append((record.id, name))
25         return result
```

Session:

```
session.py 1 X
models > session.py > ...
1  from odoo import models, fields
2
3  class Session(models.Model):
4      _name = 'clinicapablo.session'
5      _description = 'clinicapablo.session'
6
7      name = fields.Char(string="Nombre", required=True, help="Introduzca el nombre")
8      address = fields.Text(string="Descripción")
9      date = fields.Datetime(string="Fecha", required=True)
10
11     # Relacion Many2one con el modelo 'clinicapablo.treatment'
12     treatment_id = fields.Many2one("clinicapablo.treatment", string="Tratamiento", required=True, ondelete="cascade")
13
14     # Relacion Many2many con el modelo 'clinicapablo.technique'
15     technique_id = fields.Many2many("clinicapablo.technique", string="Técnicas Usadas")
16
17     # Relacion Many2one con el modelo 'clinicapablo.professional'
18     professional_id = fields.Many2one("clinicapablo.professional", string="Profesional", ondelete="cascade")
19
20     # Metodo para personalizar el nombre en listas y relaciones
21     def name_get(self):
22         result = []
23         for record in self:
24             name = f"Clinic {record.name or ''}"
25             result.append((record.id, name))
26         return result
```


Technique:

```

technique.py 1 X
models > technique.py > Technique > name_get
1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields
4  from datetime import datetime
5
6  class Technique(models.Model):
7      _name = 'clinicapablo.technique'
8      _description = 'clinicapablo.technique'
9
10     name = fields.Char(string = "Nombre", required = True, help="Introduzca el nombre")
11     address = fields.Text(string = "Descripcion")
12
13     # Relacion Many2many con el modelo 'clinicapablo.session'
14     session_id = fields.Many2many(
15         comodel_name="clinicapablo.session",
16         string="Sesiones donde se usa"
17     )
18
19     # Metodo para personalizar el nombre en listas y relaciones
20     def name_get(self):
21         result = []
22         for record in self:
23             name = f"Clinic {record.name or ''}"
24             result.append((record.id, name))
25         return result

```

Treatment:

```

treatment.py 1 X
models > treatment.py > Treatment
1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields, api
4  from datetime import datetime
5
6  class Treatment(models.Model):
7      _name = 'clinicapablo.treatment'
8      _description = 'clinicapablo.treatment'
9
10     name = fields.Char(string = "Nombre", required = True, help="Introduzca el nombre")
11     description = fields.Text(string="Descripcion")
12     date_start = fields.Datetime(string="Fecha de inicio")
13     date_end = fields.Datetime(string="Fecha de fin")
14
15     # Relacion Many2one con el modelo 'clinicapablo.register'
16     register_id = fields.Many2one("clinicapablo.register", string="Registro", required=True, ondelete="cascade")
17
18     # Relacion One2many con el modelo 'clinicapablo.treatment'
19     session_id = fields.One2many(string="Sesiones", comodel_name="clinicapablo.session", inverse_name="treatment_id")
20
21     # Relacion Many2one con el modelo 'clinicapablo.professional'
22     professional_id = fields.Many2one('clinicapablo.professional', string='Profesional', required=True)
23
24     # Metodo para personalizar el nombre en listas y relaciones
25     def name_get(self):
26         result = []
27         for record in self:
28             name = f"{record.name} - {record.register_id.name}"
29             result.append((record.id, name))
30         return result
31
32     # Validacion de fechas: la fecha de inicio debe ser anterior a la fecha de fin
33     @api.constrains('date_start', 'date_end')
34     def _check_dates(self):
35         for record in self:
36             if record.date_start and record.date_end:
37                 if record.date_start >= record.date_end:
38                     raise ValueError("La fecha de inicio debe ser anterior a la fecha de fin.")

```

3. Captura del código de la función para calcular el nombre en el modelo treatment (requisitos adicionales 1); explicándolo brevemente

El objetivo que tenemos es personalizar la forma en que se muestran los registros.

Queremos que muestre el nombre y el registro.

Ej:

(Pablo – Dolor Lumbar)

```
# Metodo para personalizar el nombre en listas y relaciones
def name_get(self):
    result = []
    for record in self:
        name = f"{record.name} - {record.register_id.name}"
        result.append((record.id, name))
    return result
```

4. Captura del código del campo computado del modelo professional que contiene las sesiones asignadas al profesional (requisitos adicionales 2); explicándolo brevemente

Este método busca todas las sesiones asignadas a un profesional y las guarda en “session_summary” cada vez que cambia treatment_id.

```
# Metodo para calcular las sesiones asignadas al profesional
@api.depends('treatment_id')
def _compute_sessions(self):
    for record in self:
        record.session_summary = self.env['clinicapablo.session'].search([('professional_id', '=', record.id)])
```

5. Captura del código del fichero xml de las vistas del modelo profesional

```
<!-- Plantilla formulario tipo form -->
<record model="ir.ui.view" id="vista_clinicapablo_professional_form">
  <field name="name">vista_clinicapablo_professional_form</field>
  <field name="model">clinicapablo.professional</field>
  <field name="arch" type="xml">
    <form string="formulario_professional" >
      <sheet>
        <group name="group_top">
          <field name="name"/>
          <field name="last_name"/>
          <field name="type"/>
          <field name="phone"/>
          <field name="email"/>
          <field name="treatment_id"/>
        </group>

        <group>
          <field name="session_summary">
            <tree>
              <field name="name"/>
              <field name="date"/>
            </tree>
          </field>
        </group>
      </sheet>
    </form>
  </field>
</record>
```

6. Captura de la vista formulario del modelo profesional de la interfaz de Odoo con los datos de un profesional y la tabla las sesiones asignadas

Nombre ? Carla

Apellidos ? Garrido Gonzalez

Tipo ? Muscular

Teléfono ?

Email ?

Tratamientos Asignados ?

Nombre	Descripcion	Fecha de inicio	Fecha de fin	Registro	Sesiones	Profesional	
Tratamiento 1	Dolor en rodilla derecha	03/04/2025 10:11:08	06/04/2025 10:11:21	Clinic Registro 1	1 registro	Clinic Carla	✕

Agregar una línea

Sesiones Asignadas ?

Nombre	Fecha
Sesion 1	04/04/2025 10:12:06

7. Captura del código del campo del modelo session para mostrar el nombre del profesional asignado a la sesión (requisitos adicionales 3); explicándolo brevemente

Personalizamos cada registro poniendo “Clinic [nombreRegistro]”.

```
# Metodo para personalizar el nombre en listas y relaciones
def name_get(self):
    result = []
    for record in self:
        name = f"Clinic {record.name or ''}"
        result.append((record.id, name))
    return result
```

8. Captura de la vista formulario del modelo session de la interfaz de Odoo con los datos de una sesión y el campo del nombre del profesional asignado

Nombre ? Sesion 1

Descripción ? Primera toma, comprobar estado de la rodilla

Fecha ? 04/04/2025 10:12:06

Tratamiento ? Tratamiento 1 - Registro 1

Técnicas Usadas ?

Nombre	Descripcion	Sesiones donde se usa	
Tecnica agujas	Usaremos las agujas para destensar los músculos. y poder trabajar correctamente	1 registro	✕

Agregar una línea

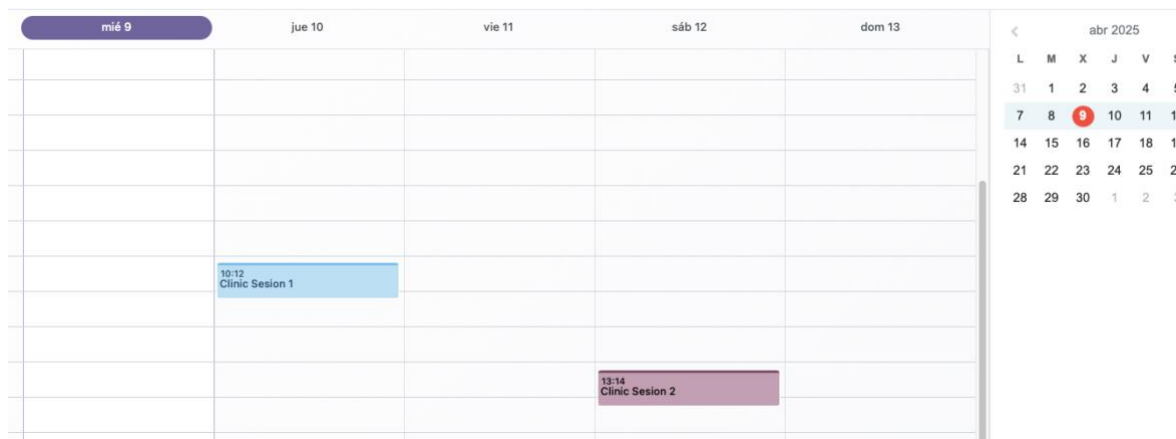
Profesional ? Clinic Carla

9. Captura del código del fichero xml de la vista calendar del modelo session (requisitos adicionales 4)

```
<!--Vista calendario-->
<record model="ir.ui.view" id="vista_clinicapablo_session_calendar">
  <field name="name">vista_clinicapablo_session_calendar</field>
  <field name="model">clinicapablo.session</field>
  <field name="arch" type="xml">
    <calendar date_start="date" color="professional_id">
      <field name="name"/>
    </calendar>
  </field>
</record>

<record model="ir.actions.act_window" id="accion_clinicapablo_session_form">
  <field name="name">Listado de sesiones</field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">clinicapablo.session</field>
  <field name="view_mode">tree,form,calendar</field>
  <field name="help" type="html">
    <p class="oe_view_nocontent_create">
      No hay sesiones registradas.
    </p>
    <p>Haz clic en <strong>'Crear'</strong> para añadir nuevas sesiones.</p>
  </field>
</record>
```

10. Captura de la vista calendar del modelo session de la interfaz de Odoo con los datos de varias sesiones de varios profesionales diferenciados por colores



11. Captura de la estructura de menús generada en la interfaz de Odoo



Herencia

Explicar el desarrollo del apartado de la herencia: objetivos, modelos utilizados, código de las vistas, vistas creadas, explicar el código de modelos y vistas, explicar el proceso seguido....

En este proyecto hemos utilizado la herencia clásica para crear los modelos necesarios para gestionar una clínica de fisioterapia. Cada modelo hereda de la clase base de Odoo y se ha personalizado con campos propios, relaciones entre entidades y restricciones para mantener la integridad de los datos. Creando además vistas y utilizando herencia de vistas para adaptar la interfaz de usuario.

Pruebas

Realizar y documentar un mínimo de 2 pruebas

Prueba 1

Creación del usuario:

/ ClinicPablo

Nombre	Descripción	Cliente	Tratamiento
Terapia manual		ClinicPablo	1 registro

Creación del tratamiento:

Tratamiento espalda - Terapia manual

Nombre	Descripción	Fecha	Tratamiento	Técnicas Usadas	Profesional
Sesion 1	Primer masaje, comprobar estado de la espalda	10/04/2025 20:39:00	Tratamiento espalda - Terapia manual	1 registro	Clinic Laura

Creación de la primera sesión:

Nombre ?	Sesion 1												
Descripción ?	Primer masaje, comprobar estado de la espalda												
Fecha ?	10/04/2025 20:39:00												
Tratamiento ?	Tratamiento espalda - Terapia manual												
Técnicas Usadas ?	<table><thead><tr><th>Nombre</th><th>Descripcion</th><th>Sesiones don</th></tr></thead><tbody><tr><td>Masaje rotativo</td><td></td><td>1 registro</td></tr><tr><td colspan="3">Agregar una línea</td></tr><tr><td colspan="3"></td></tr></tbody></table>	Nombre	Descripcion	Sesiones don	Masaje rotativo		1 registro	Agregar una línea					
Nombre	Descripcion	Sesiones don											
Masaje rotativo		1 registro											
Agregar una línea													
Profesional ?	Clinic Laura												

Creación de la fisioterapeuta:

Nombre ?	Laura												
Apellidos ?	Garrido Gonzalez												
Tipo ?	Muscular												
Teléfono ?													
Email ?													
Tratamientos Asignados ?	<table><thead><tr><th>Nombre</th><th>Descripcion</th><th>Fecha de inicio</th></tr></thead><tbody><tr><td colspan="3">Agregar una línea</td></tr><tr><td colspan="3"></td></tr><tr><td colspan="3"></td></tr></tbody></table>	Nombre	Descripcion	Fecha de inicio	Agregar una línea								
Nombre	Descripcion	Fecha de inicio											
Agregar una línea													
Sesiones Asignadas ?	<table><thead><tr><th>Nombre</th></tr></thead><tbody><tr><td></td></tr></tbody></table>	Nombre											
Nombre													

Creación del registro:

Nombre ?	Terapia manual																																
Descripcion ?																																	
Cliente ?	Clinic Pablo																																
Tratamiento ?	<table><thead><tr><th>Nombre</th><th>Descripcion</th><th>Fecha de inicio</th><th>Fecha de fin</th><th>Registro</th><th>Sesiones</th><th>Profesional</th><th></th></tr></thead><tbody><tr><td>Tratamiento espalda</td><td>Haremos unos masajes y acabaremos con agujas</td><td>10/04/2025 20:37:24</td><td>19/04/2025 20:37:24</td><td>Clinic Terapia manual</td><td>No hay registros</td><td>Clinic Laura</td><td></td></tr><tr><td colspan="8">Agregar una línea</td></tr><tr><td colspan="8"></td></tr></tbody></table>	Nombre	Descripcion	Fecha de inicio	Fecha de fin	Registro	Sesiones	Profesional		Tratamiento espalda	Haremos unos masajes y acabaremos con agujas	10/04/2025 20:37:24	19/04/2025 20:37:24	Clinic Terapia manual	No hay registros	Clinic Laura		Agregar una línea															
Nombre	Descripcion	Fecha de inicio	Fecha de fin	Registro	Sesiones	Profesional																											
Tratamiento espalda	Haremos unos masajes y acabaremos con agujas	10/04/2025 20:37:24	19/04/2025 20:37:24	Clinic Terapia manual	No hay registros	Clinic Laura																											
Agregar una línea																																	

Creación de la técnica a usar:

Nombre ?	Tecnica agujas																					
Descripción ?	Usaremos las agujas para destensar los músculos. y poder trabajar correctamente																					
Sesiones donde se usa ?																						
	<table><thead><tr><th>Nombre</th><th>Descripción</th><th>Fecha</th><th>Tratamiento</th><th>Técnicas Usadas</th><th>Profesional</th><th></th></tr></thead><tbody><tr><td>Sesion 2</td><td>tratamiento con agujas</td><td>15/04/2025 20:41:51</td><td>Tratamiento agujas - Terapia manual</td><td>1 registro</td><td>Clinic Laura</td><td>✕</td></tr><tr><td colspan="7">Agregar una línea</td></tr></tbody></table>	Nombre	Descripción	Fecha	Tratamiento	Técnicas Usadas	Profesional		Sesion 2	tratamiento con agujas	15/04/2025 20:41:51	Tratamiento agujas - Terapia manual	1 registro	Clinic Laura	✕	Agregar una línea						
Nombre	Descripción	Fecha	Tratamiento	Técnicas Usadas	Profesional																	
Sesion 2	tratamiento con agujas	15/04/2025 20:41:51	Tratamiento agujas - Terapia manual	1 registro	Clinic Laura	✕																
Agregar una línea																						

Prueba 2

En vez de hacer otro ejemplo igual, voy a capturar la pantalla y te lo voy a mostrar de diferente manera.

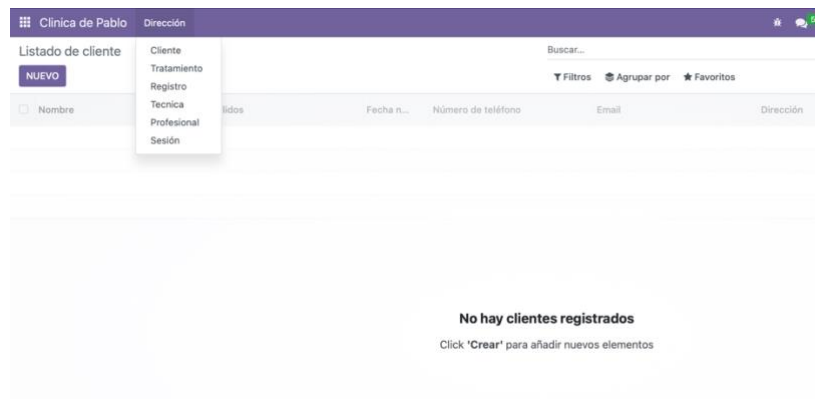
Link:

https://www.canva.com/design/DAGkK_SPSaY/AMqKf5H0ptT8GA6AMnlx_Q/edit?utm_content=DAGkK_SPSaY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

MANUAL

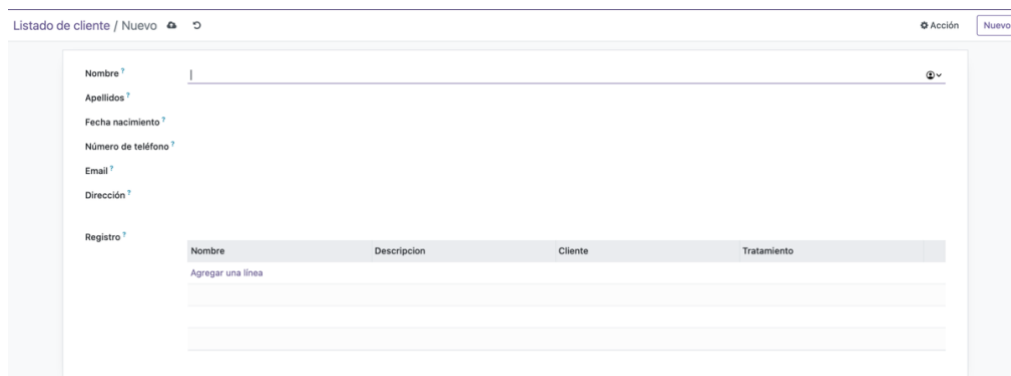
Manual de usuario:

Inicio

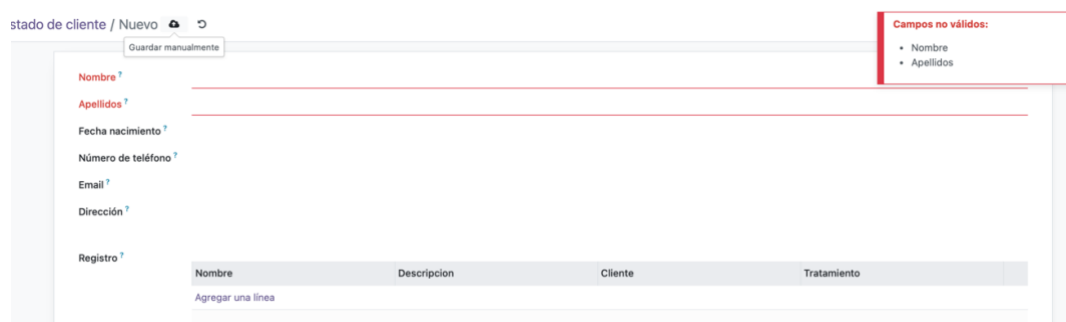


Apartados del menú

Nada más iniciar el proyecto, nos encontramos con esta primera imagen. Donde podemos registrar un cliente o donde puedes moverte a través del menú.



Si pulsas en nuevo, nos aparece lo siguiente. Lo que vemos son los campos que hemos puesto en el model.py (Client.py). Si dejásemos algún campo sin rellenar y habría que rellenarlo, nos aparecía el error avisándote de lo anterior.



Esto, nos lo encontraremos cada vez que ingresemos en una de las opciones que dispone el menú.

CONCLUSIONES Y POSIBLES AMPLIACIONES

Dificultades encontradas en el desarrollo de la aplicación, grado de satisfacción en el trabajo realizado, aprendizaje...

Posibles ampliaciones: indicar al menos una

Durante el desarrollo del módulo para la gestión de una clínica de fisioterapia en Odoo, he consolidado mis conocimientos sobre la creación de modelos, relaciones entre entidades, vistas XML y funcionalidades avanzadas como los campos computados.

Dificultades encontradas

La dificultad que más me retuvo durante el desarrollo fue la creación y personalización de la vista calendario, ya que requiere una estructura concreta y poco habitual en comparación con las vistas form y tree. Además, también supuso un reto comprender y aplicar correctamente la lógica de las funciones computadas, tanto para generar automáticamente nombres como para mostrar datos relacionados.

Grado de satisfacción

A pesar de los retos, estoy satisfecho con el resultado final del módulo. He conseguido implementar correctamente todos los requisitos, y el funcionamiento general es estable y acorde a lo esperado.

Aprendizaje

He reforzado especialmente mis habilidades en:

- La creación de relaciones entre modelos.
- El diseño estructurado de modelos con atributos y restricciones.
- El uso de campos computados y decoradores en Python.
- La organización general de un módulo en Odoo, incluyendo vistas, menús y lógica de negocio.

Implementación

- Sistema de facturación

Se ha añadido un botón en el submenú de Tratamientos en Odoo que permite crear una factura directamente desde el tratamiento seleccionado. Cuando un usuario hace clic en un tratamiento de la lista, se muestran los detalles desglosados del mismo. En la parte superior de la pantalla aparece el botón "Crear Factura". Al hacer clic en este botón, se genera automáticamente una factura con la información del tratamiento, como el precio y la descripción.

CREAR FACTURA

Nombre [?] Tratamiento 1

Descripción [?] Dolor en rodilla derecha

Fecha de inicio [?] 03/04/2025 10:11:08

Fecha de fin [?] 06/04/2025 10:11:21

Registro [?] Clinic Registro 1

Sesiones [?]

Nombre	Descripción	Fecha
Sesion 1	Primera toma, comprobar estado de la rodilla	10/04/2025 10:12:06
Sesion 2	Segunda toma, recuperar movilidad de la rodilla	12/04/2025 13:14:57

Agregar una línea

Profesional [?] Clinic Carla

Listado de tratamientos / Tratamiento 1 - Registro 1 / Borrador de factura (* 34) Imprimir Acción 1/1 < > Nuevo

CONFIRMAR **VISTA PREVIA** **CANCELAR** **BORRADOR** **PUBLICADO**

Factura de cliente

Borrador

Cliente [?] Default User Template

Fecha de factura [?]

Referencia de pago [?]

Fecha de vencimiento [?] 09/04/2025 o Términos

Líneas de factura

Otra información

Producto	Etiqueta	Cantidad	Precio	Impuestos	Subtotal	
⚙	Tratamiento 1	1,00	50,00	(IVA 21% (Bienes))	50,00 €	🗑
Agregar una línea Agregar una sección Agregar una nota						

Términos y condiciones

Base imponible: 50,00 €

IVA 21%: 10,50 €

Total: 60,50 €

BIBLIOGRAFÍA

<https://www.odoo.com/documentation>

<https://www.odoomates.tech/>