



PROYECTO SGE

CFGS Desarrollo de Aplicaciones
Multiplataforma
Informática y Comunicaciones

Memoria de proyecto managesaul

Año: 2025

Fecha de presentación: 15/01/2025

Nombre y Apellidos: Saúl Mantecón de Caso

Email: saul.mancas@educa.jcyl.es

1 Introducción.

Los sistemas de Planificación de Recursos Empresariales (ERP) son soluciones de software que integran y automatizan las funciones clave de una organización, como finanzas, recursos humanos, producción y cadena de suministro. Al centralizar la información y los procesos, los ERP facilitan la toma de decisiones y mejoran la eficiencia operativa.

Por otro lado, Scrum es un marco de trabajo ágil que ayuda a los equipos a estructurar y gestionar su trabajo mediante un conjunto de valores, principios y prácticas. Inspirado en la formación de los equipos de rugby, Scrum promueve la autoorganización y la mejora continua, permitiendo a los equipos aprender de las experiencias y adaptarse a los cambios.

La combinación de sistemas ERP con metodologías ágiles como Scrum permite a las organizaciones no solo optimizar sus procesos internos, sino también adaptarse rápidamente a las necesidades cambiantes del mercado, ofreciendo productos y servicios de mayor valor en ciclos de desarrollo más cortos

2 Organización de la memoria.

1	Introducción.	2
2	Organización de la memoria.	3
3	Estado del Arte.	4
4	Descripción general del proyecto.	8
5	Diseño de la aplicación.	9
5.1	-Base de datos:	9
5.2	-Partes del proyecto:	9
5.2.1	__init__.py	9
5.2.2	History.py.....	10
5.2.3	Project.py.....	10
5.2.4	Sprint.py.....	11
5.2.5	Task.py.	12
5.2.6	Technology.py.	13
5.2.7	Developer.py.....	13
5.2.8	Security/ir.model.acces.csv	14
5.2.9	__manifest__.py.	15
5.2.10	History.xml.	16
5.2.11	Project.xml.	18
5.2.12	Sprint.xml.	20
5.2.13	Task.xml.	22
5.2.14	Technology.py.	24
5.2.15	Developer.xml.....	25
6	Ampliación del proyecto.	27
7	Pruebas de funcionamiento.....	28
7.1	Errores de usuario.	28

7.2	Edición de registros.	29
7.3	Campos computados.	29
8	Conclusiones y posibles ampliaciones.	30
9	Github.	31
10	Bibliografía.	31

3 Estado del Arte.

ERP

a. Definición de los ERP.

Un sistema ERP ("Enterprise Resource Planning"), se traduce literalmente como "un sistema de planificación de recursos empresariales", aunque a efectos prácticos en nuestro contexto puede considerarse como un programa de gestión empresarial integrada. Son sistemas que integran todos los datos y procesos de una organización en un sistema unificado.

b. Evolución de los ERPs.

1. Nace en el 1960. Los desarrolladores informáticos del sector productivo desarrollaron programas básicos para monitorizar inventarios, balances... el origen era gestionar y controlar el inventario.
2. Entre 1970 y 1980 se desarrollan sistemas para gestionar la producción y los procesos de las empresas bajo el nombre de "Planificación de requisitos materiales" o MRP, y posteriormente llegó el sistema de planificación de recursos manufactureros o la segunda parte del MRP.
3. En **1990**, el MRP o ERP ya es funcional para controlar inventarios, producciones, gestiones administrativas y RRHH.
4. En el **2000**, Gartner declara que el concepto ERP ya es un producto terminado. Se incluía software basado en internet con acceso a tiempo real de la información y recursos de la empresa.
5. En **2006-presente**, El impulso de la nube ha cobrado gran fuerza, y ha llevado a que la herramienta ERP sea un sistema vinculado a la nube,

por otro lado, las aplicaciones han mejorado tanto en funcionalidad como en precio, actualmente ya es posible que un amplio abanico se puedan permitir este sistema que además cada vez resulta más sencillo de implantar.

c. Principales ERP

Algunos de los sistemas ERP más utilizados a nivel mundial son:

1. **SAP:** Líder global con soluciones modulares y robustas.
2. **Oracle ERP:** Enfocado en grandes empresas y aplicaciones en la nube.
3. **Microsoft Dynamics 365:** Integra ERP con herramientas de Microsoft (Office 365, Azure).
4. **Odoo:** ERP de código abierto y modular, accesible para pequeñas y medianas empresas.
5. **Infor:** Especializado en sectores industriales y de manufactura.
6. **NetSuite:** ERP basado en la nube, adquirido por Oracle.

d. ERP seleccionado (Odoo).

Es un sistema ERP de código abierto que ofrece módulos personalizables para cubrir diferentes necesidades empresariales, como ventas, inventarios, finanzas y recursos humanos. Su flexibilidad y facilidad de integración lo convierten en una opción popular tanto para pequeñas como grandes empresas.

e. Instalación y desarrollo (formas de instalación, explicando la que se va a usar para desarrollar el proyecto: Docker).

Instalación directa en el sistema operativo.

Uso de paquetes preconfigurados (DEB, RPM).

Implementación en un entorno virtual.

Contenedores Docker. Para este proyecto, se usará Docker debido a su capacidad para crear entornos aislados y replicables, facilitando el desarrollo y despliegue del sistema.

Docker es un proyecto de código abierto que automatiza el despliegue

de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

f. Especificaciones técnicas.

i. Arquitectura de Odoo.

Cliente-servidor con tres capas claramente diferenciadas:

- La base de datos en el SGBD PostgreSQL.
- El servidor Odoo contiene toda la lógica de negocio con un núcleo base y una estructura que permite ir añadiendo módulos según las necesidades de la organización. Odoo facilita una larga lista de módulos que puede ampliarse con el diseño de módulos propios para dar cobertura a las necesidades de la organización.
- La capa de clientes consiste en un cliente web, accesible desde cualquier navegador.

ii. Composición de un módulo.

Como en cualquier framework, los directorios donde se programan módulos tienen unos ficheros con unos nombres y extensiones determinados. Los módulos modifican partes de Modelo-Vista-Controlador. De esta manera, un módulo se compone de ficheros Python, XML, CSS o Javascript entre otros. Todos estos archivos deben estar en una carpeta con el nombre del módulo. Dentro de un módulo podemos encontrar:

- Ficheros Python que definen los modelos y los controladores.
- Ficheros XML que definen datos que deben ir a la base de datos. Dentro de estos datos, podemos encontrar:
 - Definición de las vistas y las acciones.
 - Datos de demo.
 - Datos estáticos que necesita el módulo.
- Ficheros estáticos como imágenes, CSS, Javascript, etc. que deben ser cargados por la interfaz web.
- Controladores web para gestionar las peticiones web. En el caso de Odoo, todo empieza con un fichero Python llamado

“__manifest__.py” que contiene la información necesaria para interpretar todos los ficheros que contiene el directorio. Esta información está almacenada usando una estructura diccionario de Python.

Además, como en cualquier paquete de Python, el directorio contiene un fichero “__init__.py”. Este tiene el nombre de los ficheros Python o directorios que contienen la lógica del módulo.

SCRUM

a. Definición de SCRUM.

Es un marco de trabajo ágil para la gestión y desarrollo de proyectos. Se enfoca en entregas incrementales y colaboración constante entre los equipos, adaptándose a los cambios y priorizando la entrega de valor al cliente.

b. Evolución.

Fue introducido en la década de 1990 por Ken Schwaber y Jeff Sutherland como una alternativa a los enfoques tradicionales de gestión de proyectos. Desde entonces, se ha convertido en uno de los marcos ágiles más utilizados en tecnología y otras industrias.

c. Funcionamiento.

El funcionamiento de SCRUM se basa en ciclos cortos de trabajo denominados sprints, durante los cuales el equipo desarrolla y entrega un incremento funcional del producto. Cada sprint incluye planificación, ejecución, revisión y retrospectiva.

d. Principales conceptos.

1. **Proyecto:** Un conjunto de actividades dirigidas a alcanzar un objetivo específico.
2. **Historias de usuario:** Descripciones breves de una funcionalidad desde la perspectiva del usuario.
3. **Sprint:** Período de tiempo fijo (generalmente 2-4 semanas) durante el cual se desarrolla un conjunto de tareas.

4. **Tarea:** Una actividad específica necesaria para completar una historia de usuario.
5. **Product Owner:** Persona responsable de definir y priorizar las funcionalidades del producto.
6. **Scrum Máster:** Facilita el proceso SCRUM, eliminando obstáculos y asegurando que el equipo siga los principios ágiles.
7. **Equipo de desarrollo:** Grupo de profesionales encargados de construir el producto.

4 Descripción general del proyecto.

El proyecto tiene como objetivo principal desarrollar un módulo para Odoo que permita gestionar proyectos de software utilizando la metodología ágil SCRUM. Este módulo incluye funcionalidades específicas para la planificación, seguimiento y ejecución de tareas relacionadas con el ciclo de vida del desarrollo ágil.

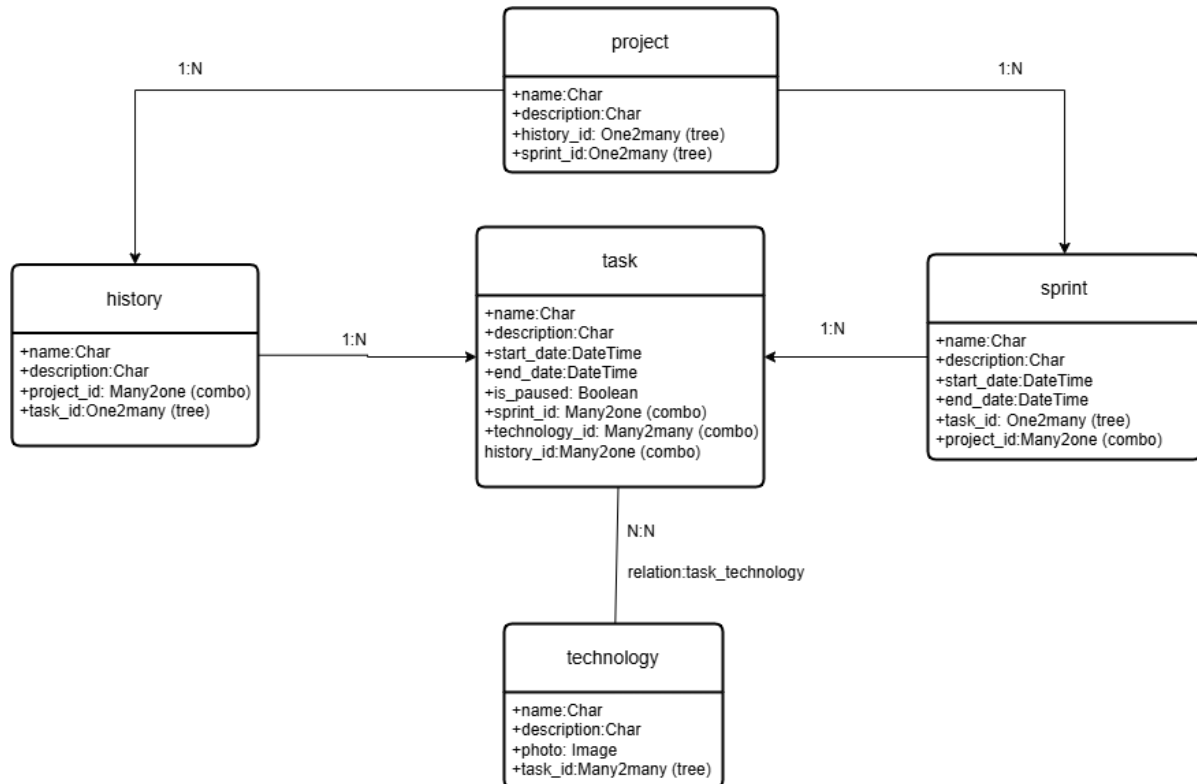
Respecto al entorno de trabajo, el proyecto se ha desarrollado en un entorno configurado con las siguientes herramientas:

1. Docker: Utilizado para la virtualización y la creación de un entorno aislado para el desarrollo del módulo.
2. PyCharm: Herramienta empleada como entorno de desarrollo integrado (IDE) para escribir, depurar y probar el código.
3. Google Chrome: Navegador web usado para realizar pruebas de la interfaz del módulo directamente en Odoo.

La estructura del proyecto tiene modelos personalizados para tareas, proyectos, sprints y tecnologías, vistas diseñadas para interactuar con los datos a través de la interfaz de Odoo y seguridad definida mediante reglas de acceso (ir.model.access.csv).

5 Diseño de la aplicación.

5.1 -Base de datos:



5.2 -Partes del proyecto:

5.2.1 __init__.py

Para poder utilizar los modelos que creamos en nuestro módulo de Odoo, es necesario indicarlo en este fichero, asegurando que dichos modelos se carguen cuando se importe el paquete `models`.

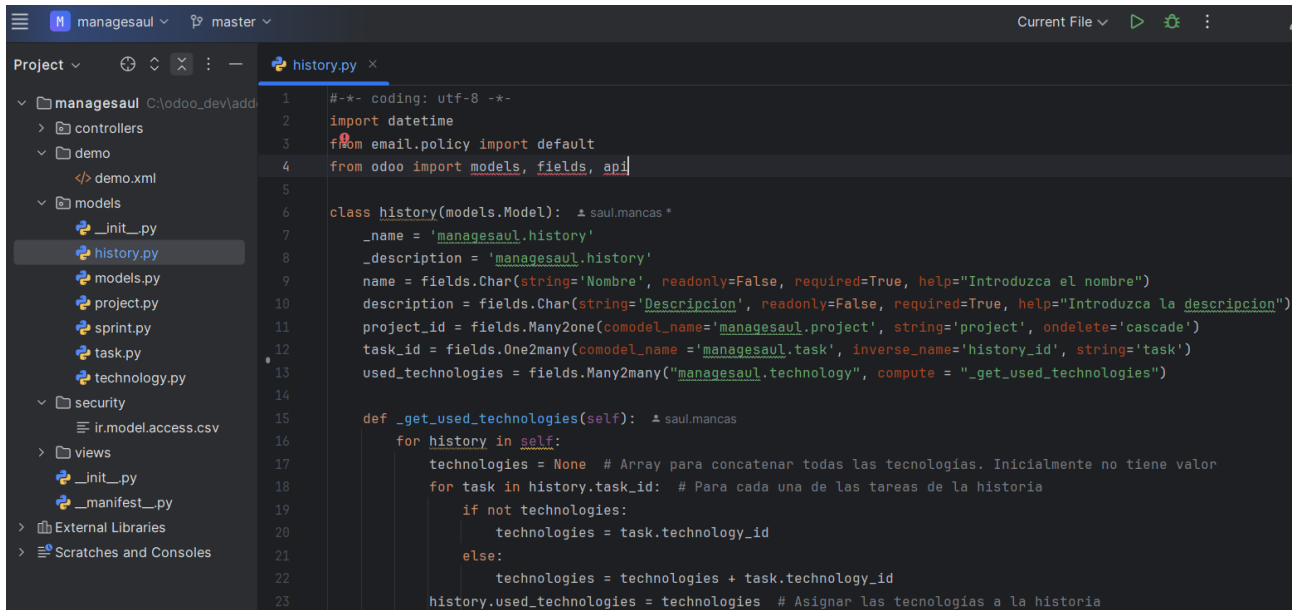
```

1  # -*- coding: utf-8 -*-
2
3  from . import models
4  from . import task
5  from . import sprint
6  from . import project
7  from . import history
8  from . import technology
9  from . import developer
    
```

5.2.2 History.py.

La clase history tiene los atributos name, description, project_id que se relaciona con proyecto con una relación 1N, task_id que se relaciona con task con una relación 1-N y used_technologies, la función _get_used_technologies calcula las tecnologías utilizadas en una historia de proyecto basándose en las tareas asociadas a esa historia: cada historia de proyecto puede tener varias tareas asociadas (relación One2many) y cada tarea puede estar vinculada a varias tecnologías (campo Many2many).

EL objetivo de la función es comprobar todas las tecnologías utilizadas en las tareas de una historia y asignarlas al campo used_technologies de esa historia.



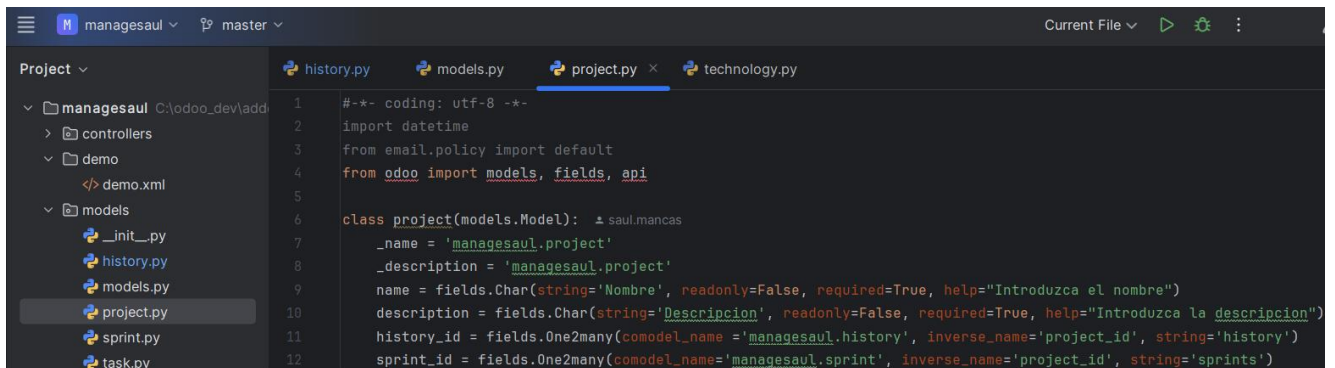
```

1  -*- coding: utf-8 -*-
2  import datetime
3  from email.policy import default
4  from odoo import models, fields, api
5
6  class history(models.Model):
7      _name = 'managesaul.history'
8      _description = 'managesaul.history'
9      name = fields.Char(string='Nombre', readonly=False, required=True, help="Introduzca el nombre")
10     description = fields.Char(string='Descripcion', readonly=False, required=True, help="Introduzca la descripcion")
11     project_id = fields.Many2one(comodel_name='managesaul.project', string='project', ondelete='cascade')
12     task_id = fields.One2many(comodel_name='managesaul.task', inverse_name='history_id', string='task')
13     used_technologies = fields.Many2many('managesaul.technology', compute='_get_used_technologies')
14
15     def _get_used_technologies(self):
16         for history in self:
17             technologies = None # Array para concatenar todas las tecnologías. Inicialmente no tiene valor
18             for task in history.task_id: # Para cada una de las tareas de la historia
19                 if not technologies:
20                     technologies = task.technology_id
21                 else:
22                     technologies = technologies + task.technology_id
23             history.used_technologies = technologies # Asignar las tecnologías a la historia

```

5.2.3 Project.py.

La clase project tiene los campos name, description, history_id relación 1-N con history y sprint_id relación 1-N con sprint.



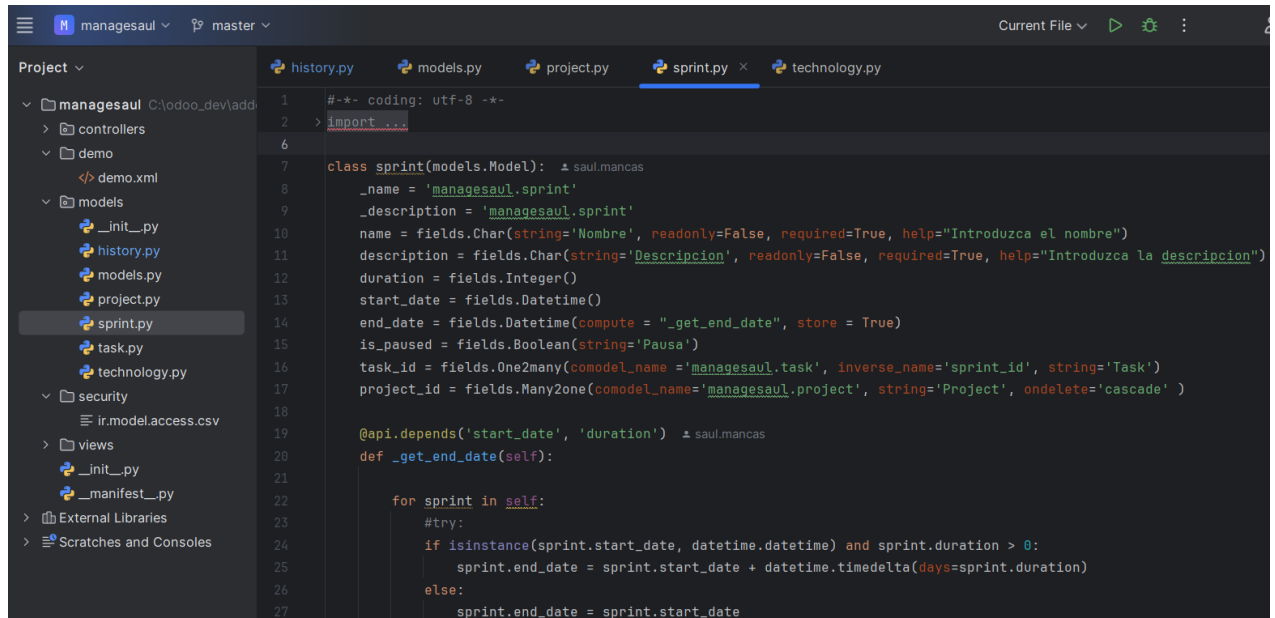
```

1  -*- coding: utf-8 -*-
2  import datetime
3  from email.policy import default
4  from odoo import models, fields, api
5
6  class project(models.Model):
7      _name = 'managesaul.project'
8      _description = 'managesaul.project'
9      name = fields.Char(string='Nombre', readonly=False, required=True, help="Introduzca el nombre")
10     description = fields.Char(string='Descripcion', readonly=False, required=True, help="Introduzca la descripcion")
11     history_id = fields.One2many(comodel_name='managesaul.history', inverse_name='project_id', string='history')
12     sprint_id = fields.One2many(comodel_name='managesaul.sprint', inverse_name='project_id', string='sprints')

```

5.2.4 Sprint.py.

La clase sprint tiene los campos name, description, duration que sirve para calcular la fecha de fin con la fecha de inicio y para ello se utiliza la función `_get_end_date`, `start_date`, `end_date`, `is_paused`, `task_id` que es una relación 1-N con task y `project_id` que tiene relación 1-N con project.

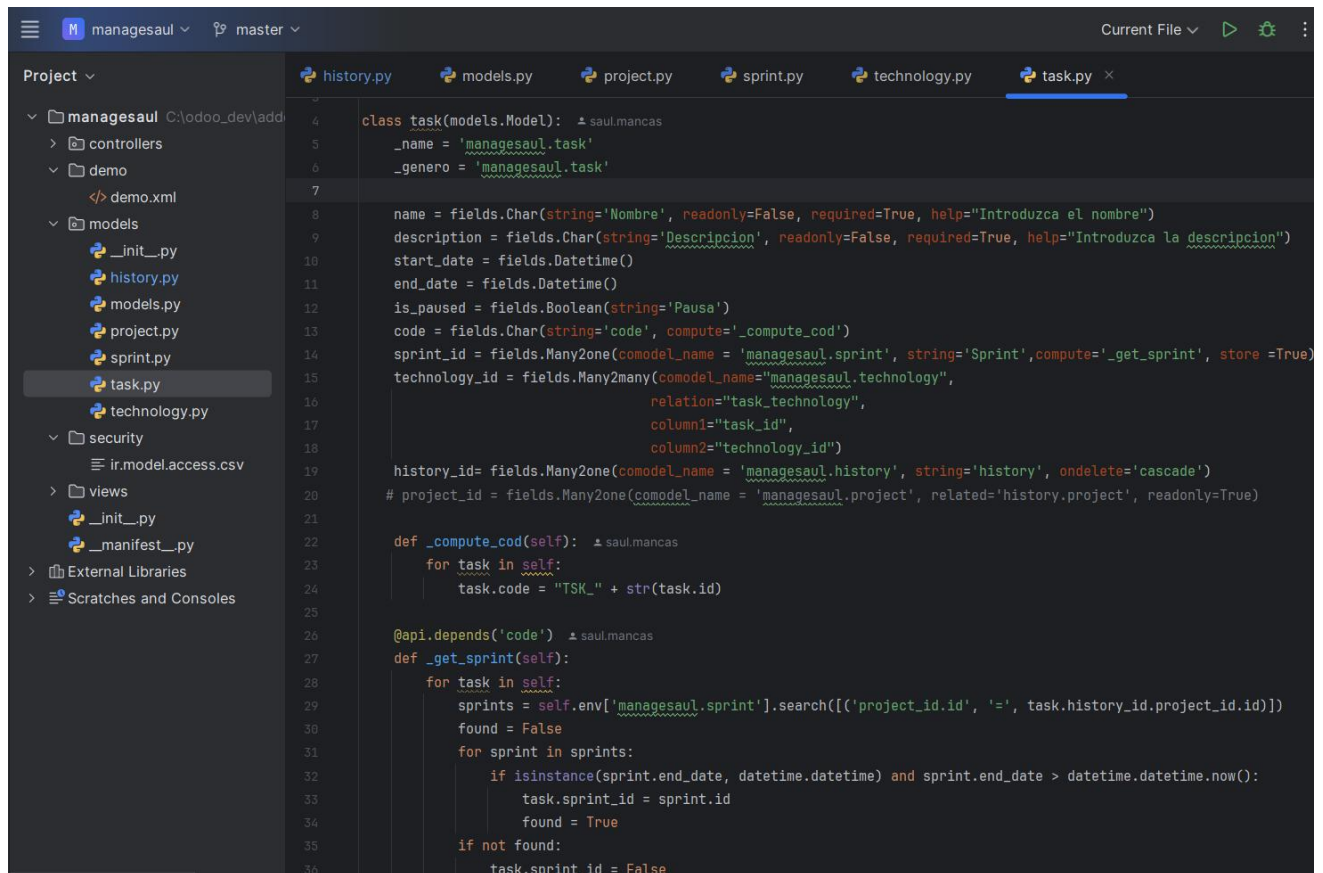


```
1  -*- coding: utf-8 -*-
2  > import ...
6
7  class sprint(models.Model):
8      _name = 'managesaul.sprint'
9      _description = 'managesaul.sprint'
10     name = fields.Char(string='Nombre', readonly=False, required=True, help="Introduzca el nombre")
11     description = fields.Char(string='Descripción', readonly=False, required=True, help="Introduzca la descripción")
12     duration = fields.Integer()
13     start_date = fields.DateTime()
14     end_date = fields.DateTime(compute = "_get_end_date", store = True)
15     is_paused = fields.Boolean(string='Pausa')
16     task_id = fields.OneToOneField(comodel_name='managesaul.task', inverse_name='sprint_id', string='Task')
17     project_id = fields.ManyToManyField(comodel_name='managesaul.project', string='Project', ondelete='cascade' )
18
19     @api.depends('start_date', 'duration')
20     def _get_end_date(self):
21
22         for sprint in self:
23             #try:
24             if isinstance(sprint.start_date, datetime.datetime) and sprint.duration > 0:
25                 sprint.end_date = sprint.start_date + datetime.timedelta(days=sprint.duration)
26             else:
27                 sprint.end_date = sprint.start_date
```

5.2.5 Task.py.

La clase task tiene los campos name, description, start_date, end_date, is_paused, code que tiene la función _compute_cod que su función es que este campo tome el siguiente valor: "TSK_" + campo id del modelo, sprint_id que es una relación 1-N de sprint. Este valor se obtiene a través de la función _get_sprint: el campo sprint se va a calcular cuando se genere el valor del campo "code" (la función _get_sprint se va a ejecutar cuando el campo code cambie), por eso se crea la función _get_sprint, con el decorador @api.depends ("code"). Technology_id que es una relación N-N con technology y por último history_id que es una relación 1-N con history.

Para continuar con el proyecto hay que seguir con project_id.



```

class task(models.Model):
    _name = 'managesaul.task'
    _genero = 'managesaul.task'

    name = fields.Char(string='Nombre', readonly=False, required=True, help="Introduzca el nombre")
    description = fields.Char(string='Descripción', readonly=False, required=True, help="Introduzca la descripción")
    start_date = fields.Datetime()
    end_date = fields.Datetime()
    is_paused = fields.Boolean(string='Pausa')
    code = fields.Char(string='code', compute='_compute_cod')
    sprint_id = fields.Many2one(comodel_name='managesaul.sprint', string='Sprint', compute='_get_sprint', store=True)
    technology_id = fields.Many2many(comodel_name='managesaul.technology',
                                   relation="task_technology",
                                   column1="task_id",
                                   column2="technology_id")
    history_id = fields.Many2one(comodel_name='managesaul.history', string='history', ondelete='cascade')
    # project_id = fields.Many2one(comodel_name='managesaul.project', related='history.project', readonly=True)

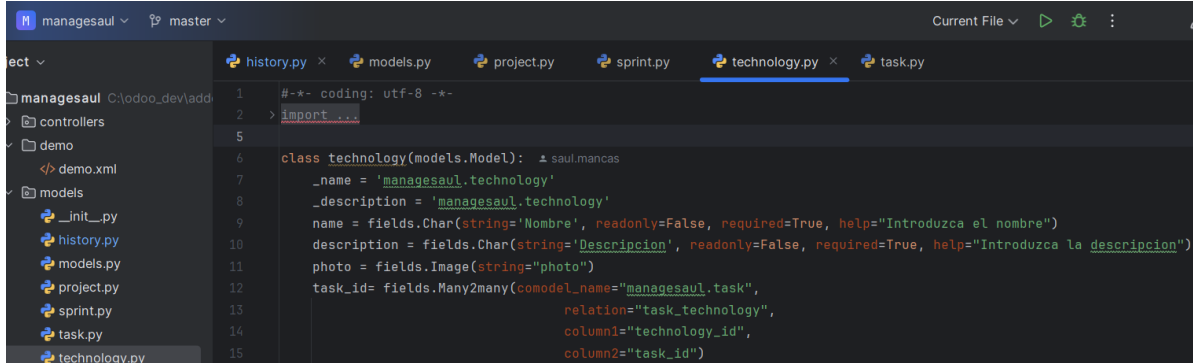
    def _compute_cod(self):
        for task in self:
            task.code = "TSK_" + str(task.id)

    @api.depends('code')
    def _get_sprint(self):
        for task in self:
            sprints = self.env['managesaul.sprint'].search([('project_id.id', '=', task.history_id.project_id.id)])
            found = False
            for sprint in sprints:
                if isinstance(sprint.end_date, datetime.datetime) and sprint.end_date > datetime.datetime.now():
                    task.sprint_id = sprint.id
                    found = True
            if not found:
                task.sprint_id = False

```

5.2.6 Technology.py.

La clase technology tiene los campos name, description, photo y task_id que es una relación N-N con task.



```

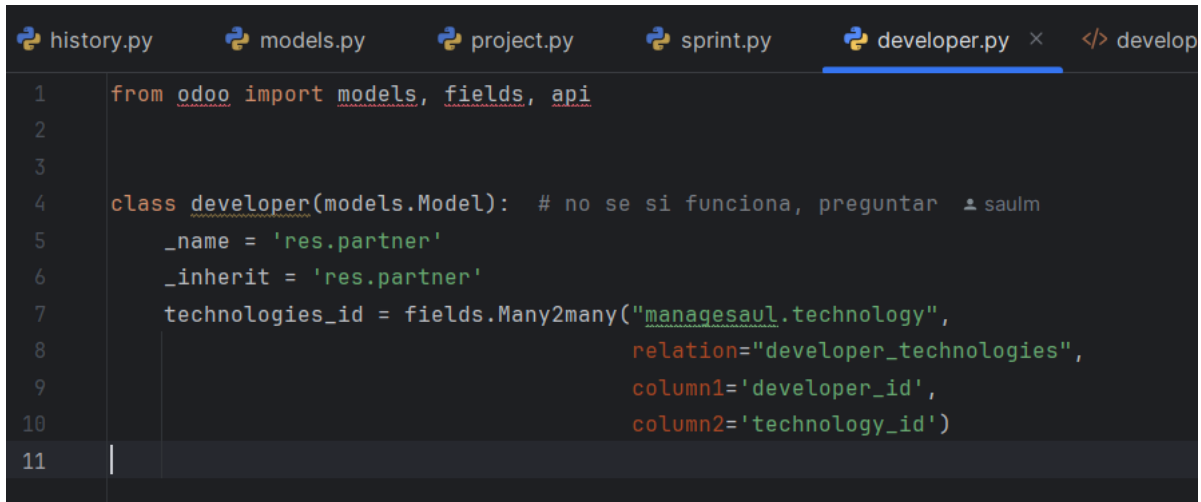
1  -*- coding: utf-8 -*-
2  > import ...
3
4
5
6  class technology(models.Model): # saul.mancas
7      _name = 'managesaul.technology'
8      _description = 'managesaul.technology'
9      name = fields.Char(string='Nombre', readonly=False, required=True, help="Introduzca el nombre")
10     description = fields.Char(string='Descripción', readonly=False, required=True, help="Introduzca la descripción")
11     photo = fields.Image(string="photo")
12     task_id = fields.Many2many(comodel_name='managesaul.task',
13                             relation="task_technology",
14                             column1="technology_id",
15                             column2="task_id")

```

5.2.7 Developer.py.

Esta clase hereda de res.partner. En este caso, manage.developer hereda la estructura de res.partner pero es un modelo independiente. El nuevo modelo tiene acceso a los campos y métodos del modelo base, pero sin modificarlo directamente. Es útil cuando se quiere reutilizar la lógica de un modelo base en un modelo completamente nuevo.

Al añadir technologies_id se añadirá este nuevo campo al modelo res.partner y no se va a crear un modelo nuevo developer:



```

1  from odoo import models, fields, api
2
3
4  class developer(models.Model): # no se si funciona, preguntar a saulm
5      _name = 'res.partner'
6      _inherit = 'res.partner'
7      technologies_id = fields.Many2many("managesaul.technology",
8                                      relation="developer_technologies",
9                                      column1='developer_id',
10                                     column2='technology_id')
11

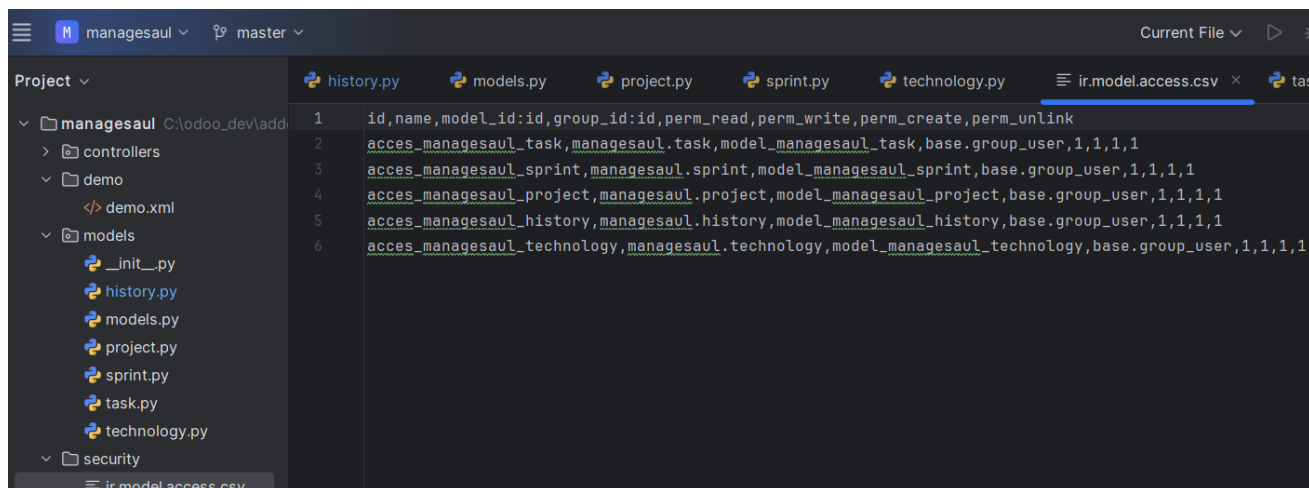
```

5.2.8 Security/ir.model.acces.csv

Este fichero contiene los permisos que se aplican a los usuarios de odoo sobre el módulo.

Este fichero está en formato CSV (Comma Separated Values o Valores Separados por Comas) y dentro de él están los campos que hacen referencia a los permisos que se aplican sobre los modelos; separados por comas.

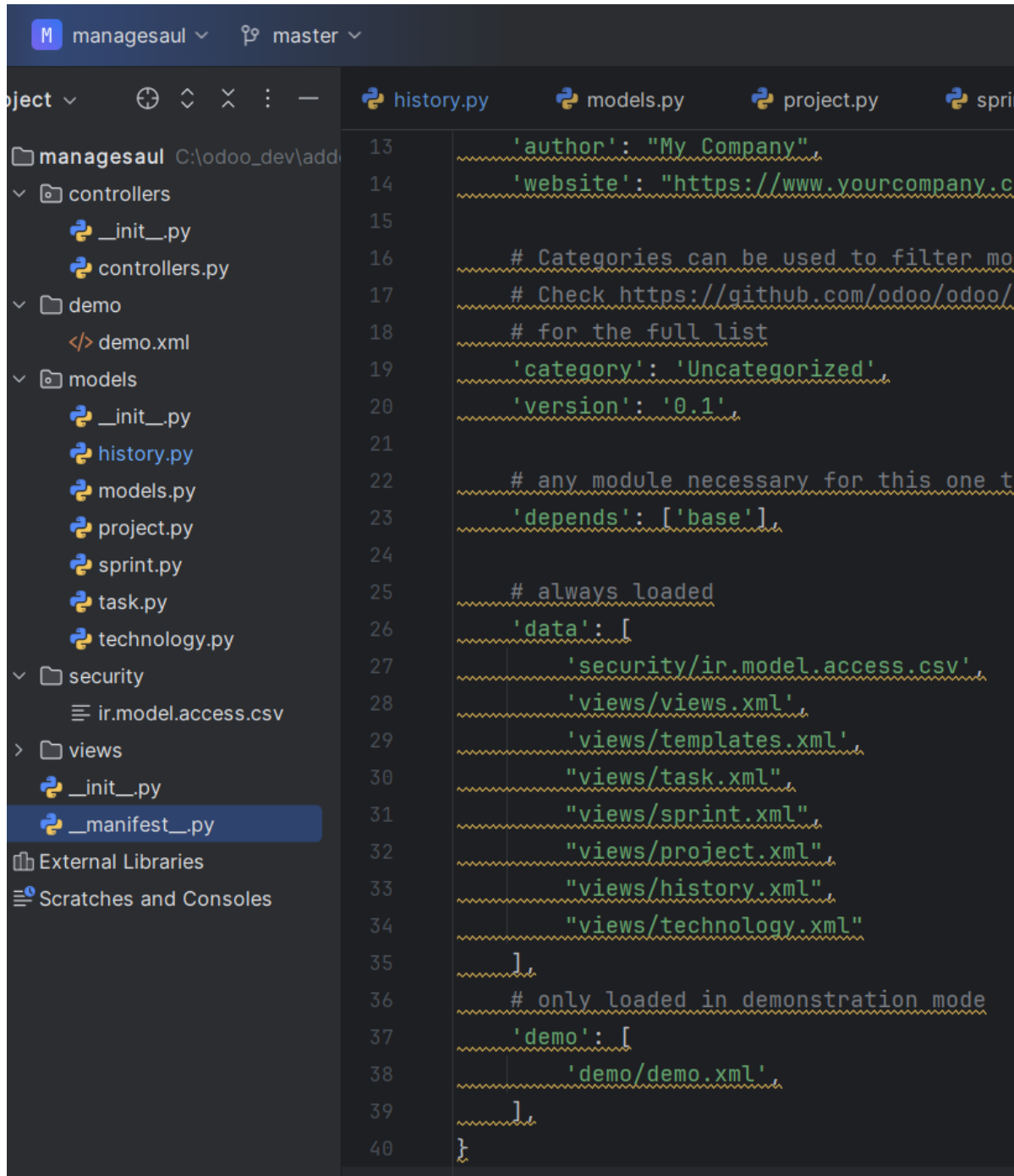
Los permisos se activan si tienen valor 1 y se desactivan si tienen valor 0.



```
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 acces_managesaul_task,managesaul.task,model_managesaul_task,base.group_user,1,1,1,1
3 acces_managesaul_sprint,managesaul.sprint,model_managesaul_sprint,base.group_user,1,1,1,1
4 acces_managesaul_project,managesaul.project,model_managesaul_project,base.group_user,1,1,1,1
5 acces_managesaul_history,managesaul.history,model_managesaul_history,base.group_user,1,1,1,1
6 acces_managesaul_technology,managesaul.technology,model_managesaul_technology,base.group_user,1,1,1,1
```

5.2.9 __manifest__.py.

Para que se vean las vistas que se mostrarán en los siguientes apartados, hace falta meterlas en este fichero en la parte de data.



The screenshot shows a code editor with a sidebar on the left displaying the project structure for 'managesaul'. The sidebar includes folders for 'controllers', 'demo', 'models', 'security', and 'views', along with various Python files. The 'views' folder is expanded, showing '__init__.py' and '__manifest__.py', which is currently selected. The main editor area displays the content of '__manifest__.py', which is a Python dictionary defining the module's metadata and data. The code includes fields for 'author', 'website', 'category', 'version', 'depends', 'data', and 'demo'.

```

13     'author': "My Company",
14     'website': "https://www.yourcompany.com",
15
16     # Categories can be used to filter modules by groups
17     # Check https://github.com/odoo/odoo/ for the full list
18
19     'category': 'Uncategorized',
20     'version': '0.1',
21
22     # any module necessary for this one to be installed
23     'depends': ['base'],
24
25     # always loaded
26     'data': [
27         'security/ir.model.access.csv',
28         'views/views.xml',
29         'views/templates.xml',
30         "views/task.xml",
31         "views/sprint.xml",
32         "views/project.xml",
33         "views/history.xml",
34         "views/technology.xml"
35     ],
36     # only loaded in demonstration mode
37     'demo': [
38         'demo/demo.xml',
39     ],
40 }

```


5.2.10 History.xml.

```

Project ▾
  ▾ managesaul C:\odoo_dev\add
    ▾ controllers
      _init_.py
      controllers.py
    ▾ demo
      demo.xml
    ▾ models
      _init_.py
      history.py
      models.py
      project.py
      sprint.py
      task.py
      technology.py
    ▾ security
      ir.model.access.csv
    ▾ views
      history.xml
      project.xml
      sprint.xml
      task.xml
      technology.xml
      templates.xml
      views.xml
      _init_.py
      _manifest_.py
  > External Libraries
  > Scratches and Consoles

models.py project.py sprint.py technology.py ir.model.a

1 <odoo>
2   <data>
3     <record model="ir.ui.view" id="vista_managesaul_history_tree">
4       <field name="name">vista_managesaul_history_tree</field>
5       <field name="model">managesaul.history</field>
6       <field name="arch" type="xml">
7         <tree>
8           <field name="name"/>
9           <field name="description"/>
10        </tree>
11      </field>
12    </record>
13
14
15    <record model="ir.ui.view" id="vista_managesaul_history_form">
16      <field name="name">vista_managesaul_history_form</field>
17      <field name="model">managesaul.history</field>
18      <field name="arch" type="xml">
19        <form string="formulario_history">
20          <sheet>
21            <group name="group_top">
22              <field name="name"/>
23              <field name="description"/>
24              <field name="project_id"/>
25              <field name="task_id"/>
26              <field name="used_technologies"/>
27            </group>
28          </sheet>
29        </form>
30      </field>
31    </record>
32  </data>
  
```

```

managesaul ▾ master ▾
Project ▾
  ▾ managesaul C:\odoo_dev\add
    ▾ controllers
      _init_.py
      controllers.py
    ▾ demo
      demo.xml
    ▾ models
      _init_.py
      history.py
      models.py
      project.py
      sprint.py
      task.py
      technology.py
    ▾ security
      ir.model.access.csv
    ▾ views
      history.xml
      project.xml
      sprint.xml
      task.xml
      technology.xml
      templates.xml
      views.xml
      _init_.py
      _manifest_.py
  > External Libraries
  > Scratches and Consoles

models.py project.py sprint.py technology.py ir.model.access.csv _init_.py _m

1 <odoo>
2   <data>
3     <!-- Plantilla action -->
4     <record model="ir.actions.act_window" id="accion_managesaul_history_form">
5       <field name="name">Listado de history</field>
6       <field name="type">ir.actions.act_window</field>
7       <field name="res_model">managesaul.history</field>
8       <field name="view_mode">tree,form</field>
9       <field name="help" type="html">
10        <p class="oe_view_nocontent_create">
11          history
12        </p>
13        <p>Click <strong>'Crear'</strong> para añadir nuevos elementos
14        </p>
15      </field>
16    </record>
17
18    <!-- Top menu item -->
19    <!-- menu categories -->
20
21    <menuitem name="Management de saul" id="managesaul_management" parent="managesaul.manage_root"/>
22    <menuitem name="history" id="managesaul_history" parent="managesaul_management"
23      action="accion_managesaul_history_form"/>
24
25    <!-- actions -->
26
27    <!--<menuitem name="Server to list" id="filmotecasaul" parent="filmotecasaul.menu_2"
28      action="filmotecasaul.action_server"/>
29    -->
30  </data>
31 </odoo>
  
```


Manage de saul Management de saul 5 2 YourCompany Mitchell Admin (sgepruebas)

Listado de history / Nuevo Acción Nuevo

Nombre? history1

Descripcion? history1

project?

task?

Nombre	Descripcion	code
Añadir una línea		

Used Technologies?

Nombre	Descripcion	photo

Manage de saul Management de saul 5 2 YourCompany Mitchell Admin (sgepruebas)

Listado de history NUEVO Task Sprint project history technology

Buscar...

Filtros Agrupar por Favoritos 1-1 / 1 < >

Nombre	Descripcion
<input type="checkbox"/> history1	history1

Se observa que funciona la implementación de used_technologies.

Manage de saul Management de saul 5 2 YourCompany Mitchell Admin

Listado de history / history1 Acción 1 / 1 < >

Nombre? history1

Descripcion? history1

project?

task?

Nombre	Descripcion	code
Task1	Task1	TSK_2
Añadir una línea		

Used Technologies?

Nombre	Descripcion	photo
tech1	tech1	

5.2.11 Project.xml.

```

1 <!-- Project -->
2 <record model="ir.ui.view" id="vista_managesaul_project_tree">
3   <field name="name">vista_managesaul_project_tree</field>
4   <field name="model">managesaul.project</field>
5   <field name="arch" type="xml">
6     <tree>
7       <field name="name"/>
8       <field name="description"/>
9     </tree>
10  </field>
11 </record>
12
13 <record model="ir.ui.view" id="vista_managesaul_project_form">
14   <field name="name">vista_managesaul_project_form</field>
15   <field name="model">managesaul.project</field>
16   <field name="arch" type="xml">
17     <form string="formulario_project">
18       <sheet>
19         <group name="group_top">
20           <field name="name"/>
21           <field name="description"/>
22           <field name="history_id"/>
23           <field name="sprint_id"/>
24         </group>
25       </sheet>
26     </form>
27   </field>
28 </record>
29
30 </data>
31

```

```

<!-- Plantilla action -->
<record model="ir.actions.act_window" id="accion_managesaul_project_form">
  <field name="name">Listado de project</field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">managesaul.project</field>
  <field name="view_mode">tree,form</field>
  <field name="help" type="html">
    <p class="oe_view_nocontent_create">
      project
    </p>
    <p>Click <strong>'Crear'</strong> para añadir nuevos elementos</p>
  </field>
</record>

<!-- Top menu item -->
<!-- menu categories -->

<menuitem name="Management de saul" id="managesaul_management" parent="managesaul.manage_root"/>
<menuitem name="project" id="managesaul_project" parent="managesaul_management"
  action="accion_managesaul_project_form"/>
<!-- actions -->

<!--<menuitem name="Server to list" id="filmotecasaul" parent="filmotecasaul.menu_2"
  action="filmotecasaul.action_server"/>
-->

</data>
</odoo>

```

Manage de saul | Management de saul

5 2 YourCompany Mitchell Admin

Listado de project / Project1

Acción 1 / 1

Nombre ? Project1

Descripcion ? Project1

history ?

Nombre	Descripcion
Añadir una línea	

sprints ?

Nombre	Descripcion
Añadir una línea	

Manage de saul | Management de saul

5 2 YourCompany Mitchell Admin (sigpruebas)

Listado de project

Buscar...

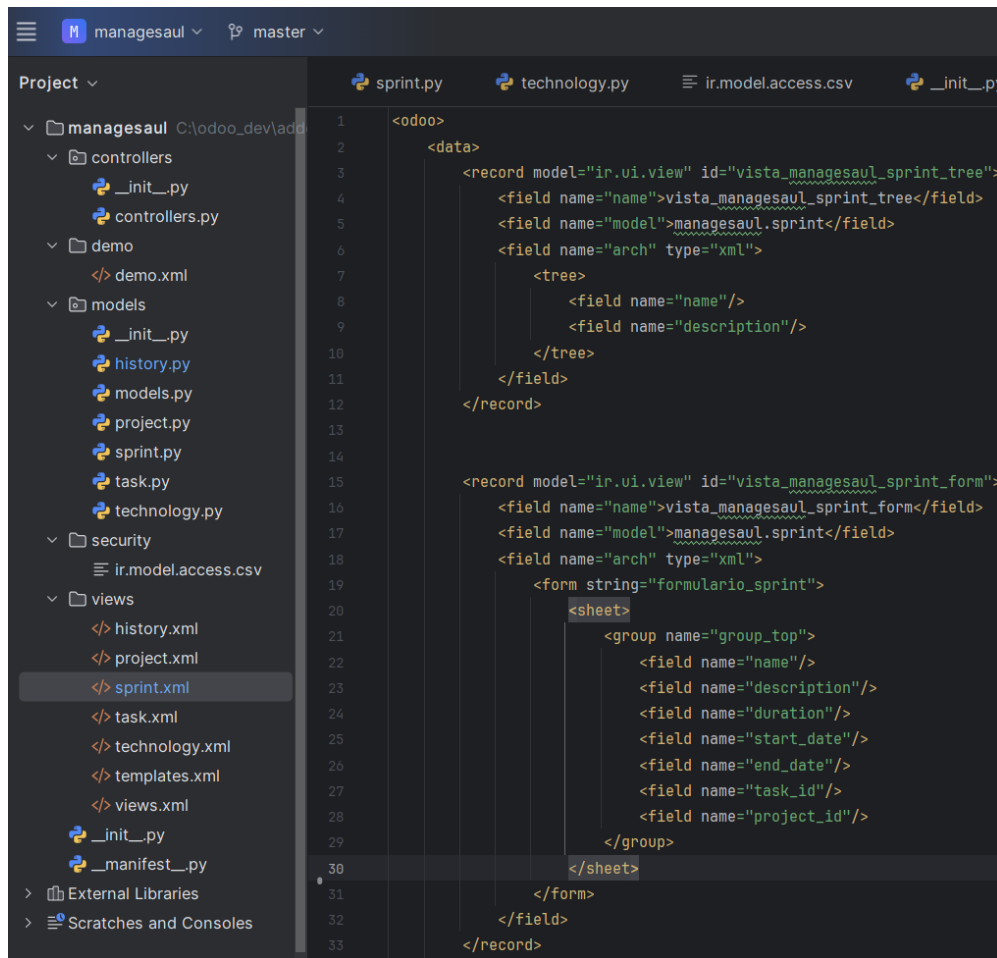
NUEVO

Filtros Agrupar por Favoritos

1-1 / 1

Nombre	Descripcion
Project1	Project1

5.2.12 Sprint.xml.



The screenshot shows an IDE with a project named 'managesaul'. The file explorer on the left shows the project structure, including folders for controllers, demo, models, security, and views. The 'views' folder is expanded, showing files like history.xml, project.xml, **sprint.xml**, task.xml, technology.xml, templates.xml, and views.xml. The 'sprint.xml' file is selected and its content is displayed in the main editor. The XML content is as follows:

```

1 <odoo>
2   <data>
3     <record model="ir.ui.view" id="vista_managesaul_sprint_tree">
4       <field name="name">vista_managesaul_sprint_tree</field>
5       <field name="model">managesaul.sprint</field>
6       <field name="arch" type="xml">
7         <tree>
8           <field name="name"/>
9           <field name="description"/>
10        </tree>
11      </field>
12    </record>
13
14    <record model="ir.ui.view" id="vista_managesaul_sprint_form">
15      <field name="name">vista_managesaul_sprint_form</field>
16      <field name="model">managesaul.sprint</field>
17      <field name="arch" type="xml">
18        <form string="formulario_sprint">
19          <sheet>
20            <group name="group_top">
21              <field name="name"/>
22              <field name="description"/>
23              <field name="duration"/>
24              <field name="start_date"/>
25              <field name="end_date"/>
26              <field name="task_id"/>
27              <field name="project_id"/>
28            </group>
29          </sheet>
30        </form>
31      </field>
32    </record>
33  </data>
34 </odoo>

```

```

<!-- Plantilla action -->
<record model="ir.actions.act_window" id="accion_managesaul_sprint_form">
  <field name="name">Listado de sprint</field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">managesaul.sprint</field>
  <field name="view_mode">tree,form</field>
  <field name="help" type="html">
    <p class="oe_view_nocontent_create">
      Sprint
    </p>
    <p>Click <strong>'Crear'</strong> para añadir nuevos elementos</p>
  </field>
</record>

<!-- Top menu item -->
<!-- menu categories -->

<menuitem name="Sprint" id="managesaul_sprint" parent="managesaul_management"
  action="accion_managesaul_sprint_form"/>

<!-- actions -->

<!--<menuitem name="Server to list" id="filmotecasaul" parent="filmotecasaul.menu_2"
  action="filmotecasaul.action_server"/>
-->

</data>
</odoo>

```

Manage de saul

Management de saul

5

2

YourCompany

Mitchell Admin (sgepruebas)

Listado de sprint / sprint1

Acción

1 / 1

Nuevo

Nombre ?

sprint1

Description ?

sprint1

Duration ?

3

Start Date ?

22/12/2024 01:13:13

End Date ?

25/12/2024 01:13:13

Task ?

Nombre	Description	code
Añadir una línea		

Project ?

Project1

Manage de saul

Management de saul

5

2

YourCompany

Mitchell Admin (sgepruebas)

Listado de sprint

NUEVO

Buscar...

Filtros

Agrupar por

Favoritos

1-1 / 1

☐

Nombre

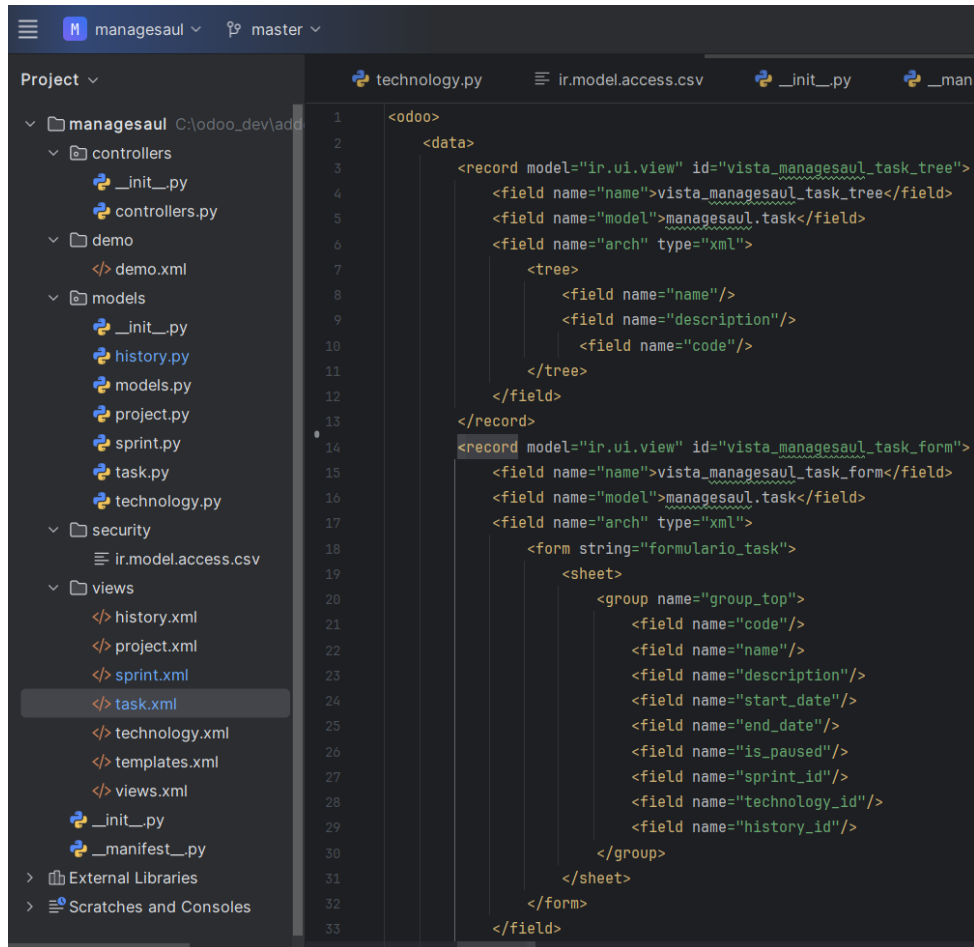
☐

sprint1

Description

sprint1

5.2.13 Task.xml.



```

1 <odoo>
2 <data>
3 <record model="ir.ui.view" id="vista_managesaul_task_tree">
4 <field name="name">vista_managesaul_task_tree</field>
5 <field name="model">managesaul.task</field>
6 <field name="arch" type="xml">
7 <tree>
8 <field name="name"/>
9 <field name="description"/>
10 <field name="code"/>
11 </tree>
12 </field>
13 </record>
14 <record model="ir.ui.view" id="vista_managesaul_task_form">
15 <field name="name">vista_managesaul_task_form</field>
16 <field name="model">managesaul.task</field>
17 <field name="arch" type="xml">
18 <form string="formulario_task">
19 <sheet>
20 <group name="group_top">
21 <field name="code"/>
22 <field name="name"/>
23 <field name="description"/>
24 <field name="start_date"/>
25 <field name="end_date"/>
26 <field name="is_paused"/>
27 <field name="sprint_id"/>
28 <field name="technology_id"/>
29 <field name="history_id"/>
30 </group>
31 </sheet>
32 </form>
33 </field>

```

```

<!-- Plantilla action -->
<record model="ir.actions.act_window" id="accion_managesaul_task_form">
  <field name="name">Listado de task</field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">managesaul.task</field>
  <field name="view_mode">tree,form</field>
  <field name="help" type="html">
    <p class="oe_view_nocontent_create">
      Task
    </p>
    <p>Click <strong>'Crear'</strong> para añadir nuevos elementos</p>
  </field>
</record>

<!-- Top menu item -->
<!-- menu categories -->
<menuitem name="Manage de saul" id="managesaul.manage_root"/>

<menuitem name="Management de saul" id="managesaul_management" parent="managesaul.manage_root"/>
<menuitem name="Task" id="managesaul_task" parent="managesaul_management"
  action="accion_managesaul_task_form"/>
<!-- actions -->

<!--<menuitem name="Server to list" id="filmotecasaul" parent="filmotecasaul.menu_2"
  action="filmotecasaul.action_server"/>
-->

```

Manage de saul
Management de saul

5
2
YourCompany
Mitchell Admin (sigepuebas)

Listado de task / Task1

Acción
1 / 1
Nuevo

code?
TSK_2

Nombre?
Task1

Descripcion?
Task1

Start Date?
22/12/2024 01:16:14

End Date?
22/12/2024 01:16:14

Pausa?
☒

Sprint?

Technology?

Nombre	Descripcion	photo
tech1	tech1	
Añadir una línea		

history?
history1

Manage de saul
Management de saul

5
2
YourCompany
Mitchell Admin (sigepuebas)

Listado de task

NUEVO

Filtros
Agrupar por
Favoritos

1-2 / 2

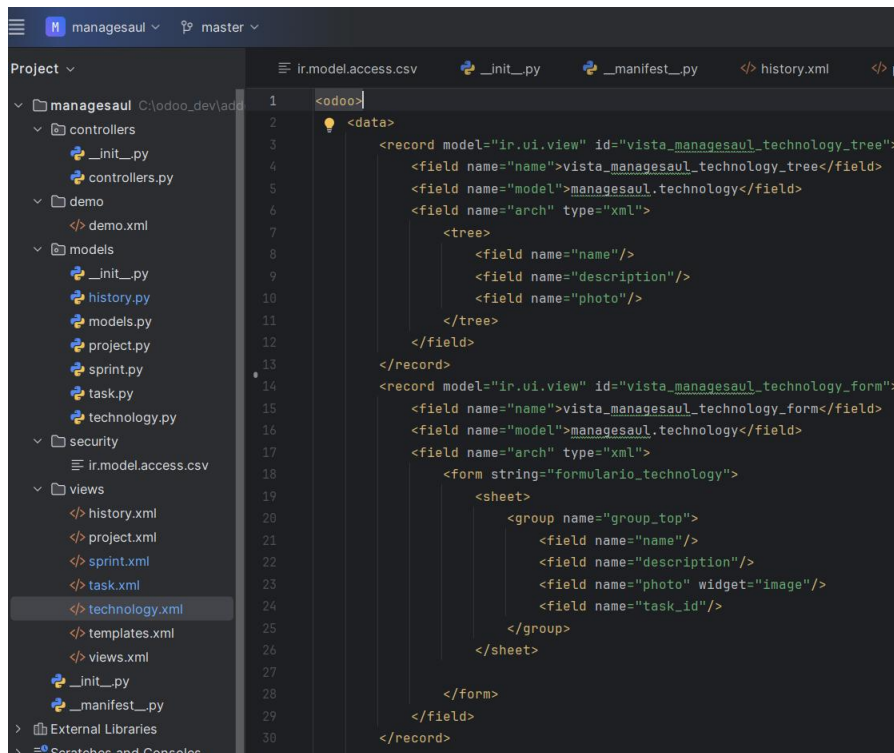
☐ Nombre

☐ task

☐ Task1

Nombre	Descripcion	code
task	task	TSK_1
Task1	Task1	TSK_2

5.2.14 Technology.py.



The screenshot shows an IDE with a project named 'managesaul'. The left sidebar displays the project structure, including folders for controllers, demo, models, security, and views. The 'views' folder is expanded, showing files like history.xml, project.xml, sprint.xml, task.xml, technology.xml, templates.xml, and views.xml. The 'technology.xml' file is selected and its content is displayed in the main editor. The XML content defines two views: 'vista_managesaul_technology_tree' and 'vista_managesaul_technology_form'.

```

1 <?xml version="1.0"?>
2 <data>
3   <record model="ir.ui.view" id="vista_managesaul_technology_tree">
4     <field name="name">vista_managesaul_technology_tree</field>
5     <field name="model">managesaul.technology</field>
6     <field name="arch" type="xml">
7       <tree>
8         <field name="name"/>
9         <field name="description"/>
10        <field name="photo"/>
11      </tree>
12    </field>
13  </record>
14  <record model="ir.ui.view" id="vista_managesaul_technology_form">
15    <field name="name">vista_managesaul_technology_form</field>
16    <field name="model">managesaul.technology</field>
17    <field name="arch" type="xml">
18      <form string="formulario_technology">
19        <sheet>
20          <group name="group_top">
21            <field name="name"/>
22            <field name="description"/>
23            <field name="photo" widget="image"/>
24            <field name="task_id"/>
25          </group>
26        </sheet>
27      </form>
28    </field>
29  </record>

```

```

<!-- Plantilla action -->
<record model="ir.actions.act_window" id="accion_managesaul_technology_form">
  <field name="name">Listado de technology</field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">managesaul.technology</field>
  <field name="view_mode">tree,form</field>
  <field name="help" type="html">
    <p class="oe_view_nocontent_create">
      technology
    </p>
    <p>Click <strong>'Crear'</strong> para añadir nuevos elementos</p>
  </field>
</record>

<!-- Top menu item -->
<!-- menu categories -->

<menuitem name="Management de saul" id="managesaul_management" parent="managesaul.manage_root"/>
<menuitem name="technology" id="managesaul_technology" parent="managesaul_management"
  action="accion_managesaul_technology_form"/>
<!-- actions -->


<!--<menuitem name="Server to list" id="filmotecasaul" parent="filmotecasaul.menu_2"
  action="filmotecasaul.action_server"/>
-->

```


Manage de saul Management de saul

Listado de technology / technolgy1

Nombre? technolgy1
Description? technolgy1
photo?



Task?

Nombre	Description	code
task	task	TSK_1

Añadir una línea

Manage de saul Management de saul

Listado de technology

Buscar...

NUEVO

Filtros Agrupar por Favoritos 1-2 / 2

Nombre	Description	photo
tech1	tech1	
technolgy1	technolgy1	

5.2.15 Developer.xml.

managesaul master

Current File

object

- controllers.py
- demo
 - demo.xml
- models
 - __init__.py
 - developer.py
 - history.py
 - models.py
 - project.py
 - sprint.py
 - task.py
 - technology.py
- security
 - ir.model.access.csv
- views
 - developer.xml
 - history.xml
 - project.xml
 - sprint.xml
 - task.xml
 - technology.xml
 - templates.xml
 - views.xml
- __init__.py
- manifest.py

```

1 <odoo>
2
3 <data>
4 <!-- Herencia de vista -->
5 <record model="ir.ui.view" id="manage_devs_partner_form">
6   <field name="name">Manage Devs Form</field>
7   <field name="model">res.partner</field>
8   <field name="inherit_id" ref="base.view_partner_form"/> <!-- Herencia -->
9   <field name="arch" type="xml">
10     <xpath expr="//sheet" position="inside"> <!-- esto no hace falta -->
11       <group>
12         <field name="technologies_id"/>
13       </group>
14     </xpath>
15   </field>
16 </record>
17
18 <!-- Acción -->
19 <record model="ir.actions.act_window" id="action_manage_devs">
20   <field name="name">Manage Developers</field>
21   <field name="res_model">res.partner</field>
22   <field name="view_mode">tree,form</field>
23 </record>
24
25 <!-- Menú -->
26 <menuitem name="Developers" id="menu_manage_devs" parent="managesaul_management" action="action_manage_devs"/>
27 </data>
28 </odoo>
  
```

Se puede observar en la imagen el botón Developer en el menú Management de saul. En la vista tree aparecen todos los contactos.

Manage de saul		Management de saul						
Manage Developers		Buscar...						
NUEVO		Filtros Agrupar por Favoritos						
Nombre		Teléfono	Correo electrónico	Comercial	Actividades	Ciudad	País	Compañía
<input type="checkbox"/>	Azure Interior	(870)-931-0505	azure.interior24@example.com		○	Fremont	Estados Unidos	
<input type="checkbox"/>	Azure Interior, Brandon Freeman	(355)-687-3262	brandon.freeman55@example.com		○	Fremont	Estados Unidos	
<input type="checkbox"/>	Azure Interior, Colleen Diaz	(255)-595-8393	colleen.diaz83@example.com		○	Fremont	Estados Unidos	
<input type="checkbox"/>	Azure Interior, Nicole Ford	(946)-638-6034	nicole.ford75@example.com		○	Fremont	Estados Unidos	
<input type="checkbox"/>	Deco Addict	(603)-996-3829	deco.addict82@example.com		○	Pleasant Hill	Estados Unidos	
<input type="checkbox"/>	Deco Addict, Addison Olson	(223)-399-7637	addison.olson28@example.com		○	Pleasant Hill	Estados Unidos	
<input type="checkbox"/>	Deco Addict, Douglas Fletcher	(132)-553-7242	douglas.fletcher51@example.com		○	Pleasant Hill	Estados Unidos	
<input type="checkbox"/>	Deco Addict, Floyd Steward	(145)-138-3401	floyd.steward34@example.com		○	Pleasant Hill	Estados Unidos	
<input type="checkbox"/>	ES Company	+34 612 34 56 78	info@company.es@example.com		○	Candanos	España	
<input type="checkbox"/>	Gemini Furniture	(941)-284-4875	gemini.furniture39@example.com		○	Fairfield	Estados Unidos	
<input type="checkbox"/>	Gemini Furniture, Edwin Hansen	(943)-352-2555	edwin.hansen58@example.com		○	Fairfield	Estados Unidos	
<input type="checkbox"/>	Gemini Furniture, Jesse Brown	(829)-386-3277	jesse.brown74@example.com		○	Fairfield	Estados Unidos	YourCompany
<input type="checkbox"/>	Gemini Furniture, Oscar Morgan	(561)-239-1744	oscar.morgan11@example.com		○	Fairfield	Estados Unidos	
<input type="checkbox"/>	Gemini Furniture, Soham Palmer	(379)-167-2040	soham.palmer15@example.com		○	Fairfield	Estados Unidos	
<input type="checkbox"/>	Lumber Inc	(828)-316-0593	lumber-inc92@example.com		○	Stockton	Estados Unidos	
<input type="checkbox"/>	Lumber Inc, Lorraine Douglas	(443)-648-9155	lorraine.douglas35@example.com		○	Stockton	Estados Unidos	
<input type="checkbox"/>	Readv Mat	(803)-873-6126	readv.mat28@example.com		○	Tracy	Estados Unidos	

En el formulario form podemos observar como aparece el campo que technologies que se mencionó anteriormente.

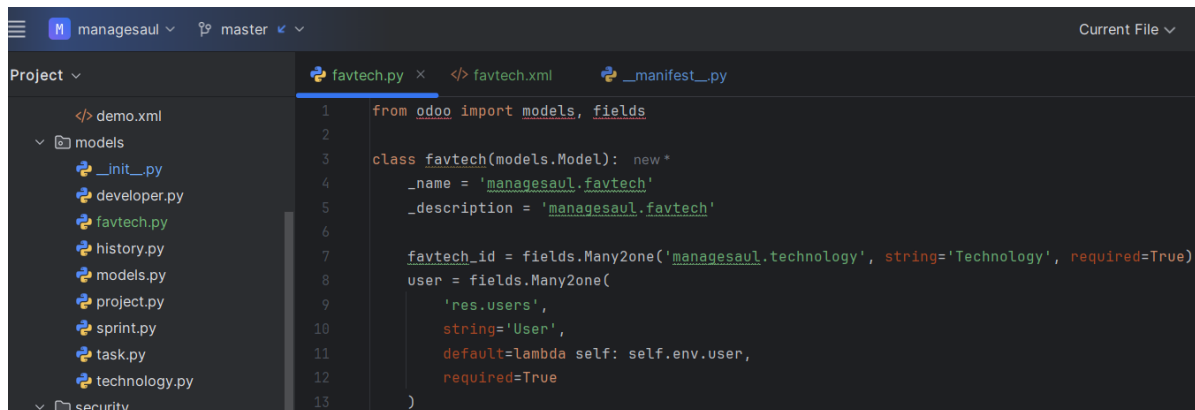
Manage de saul		Management de saul						
Manage Developers / Nuevo		Acción Nuevo						
Individuo Compañía		Enviar mensaje Registrar una nota Actividades 0 Seguir						
p. ej. Brandom Freeman		Hoy Mitchell Admin Creando un nuevo registro...						
Nombre de la empresa...								
Contacto		Puesto de trabajo ? p. ej. director de ventas						
Calle...		Teléfono ?						
Calle 2...		Móvil ?						
Provincia C.P. Ciudad		Correo electrónico ?						
País		Sitio web ? p. ej. https://www.odoo.com						
NIF ? p. ej. ESA00000000		Título ? p. ej. Señor						
		Etiquetas ? p. ej. "B2B", "VIP", "consultoría", ...						
Contactos y direcciones Ventas y compras Facturación / Contabilidad Notas internas								
AÑADIR								
Technologies ?								
Nombre Descripción photo								
Añadir una línea								

6 Ampliación del proyecto.

He implementado el apartado “Tecnologías favoritas” para ofrecer a los usuarios una herramienta que les permita registrar y consultar las tecnologías que consideran más relevantes o útiles dentro del sistema.

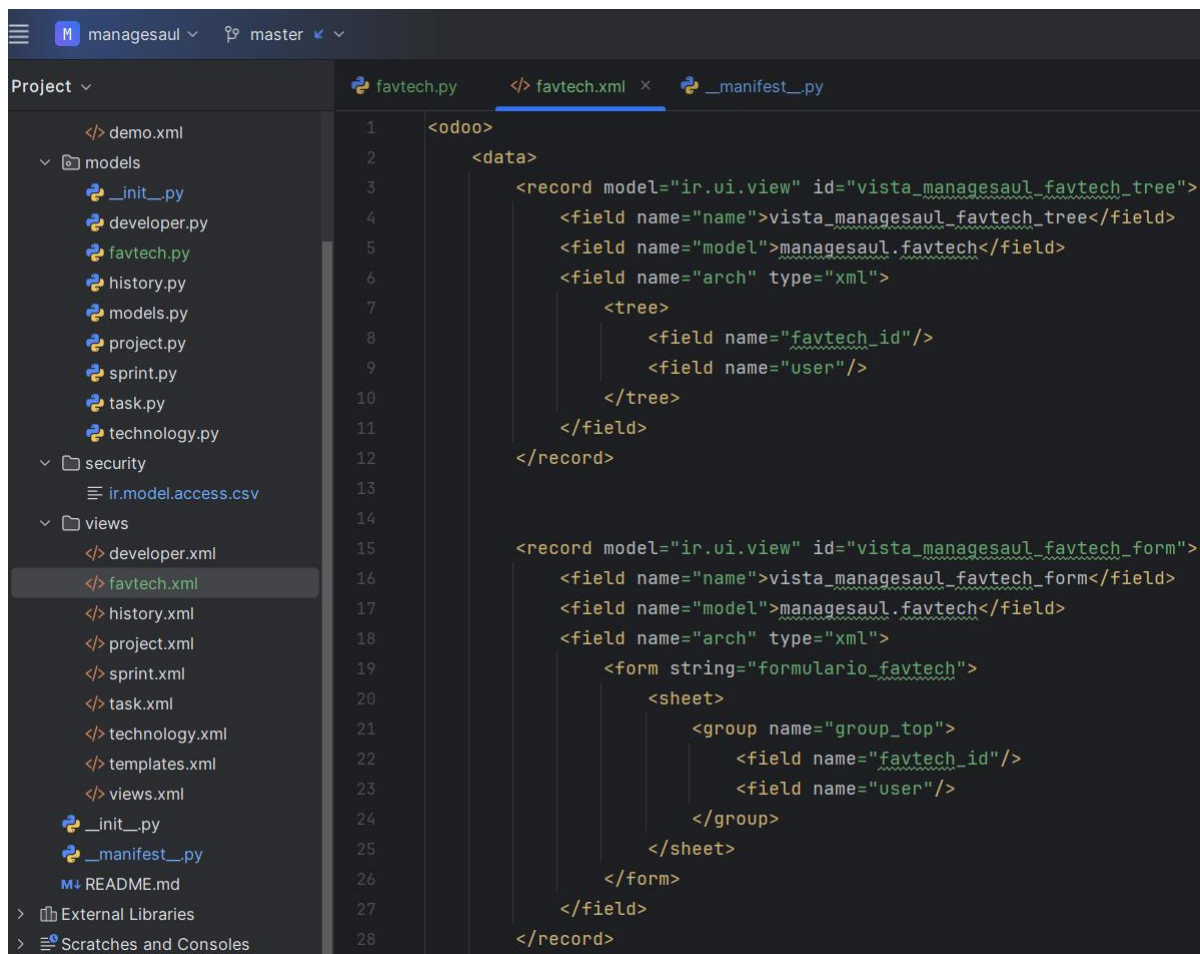
Para ello he creado el modelo favtech junto a su vista.

Favtech_id está relacionado con el technology y user está directamente relacionado al usuario que está conectado en la sesión.



```

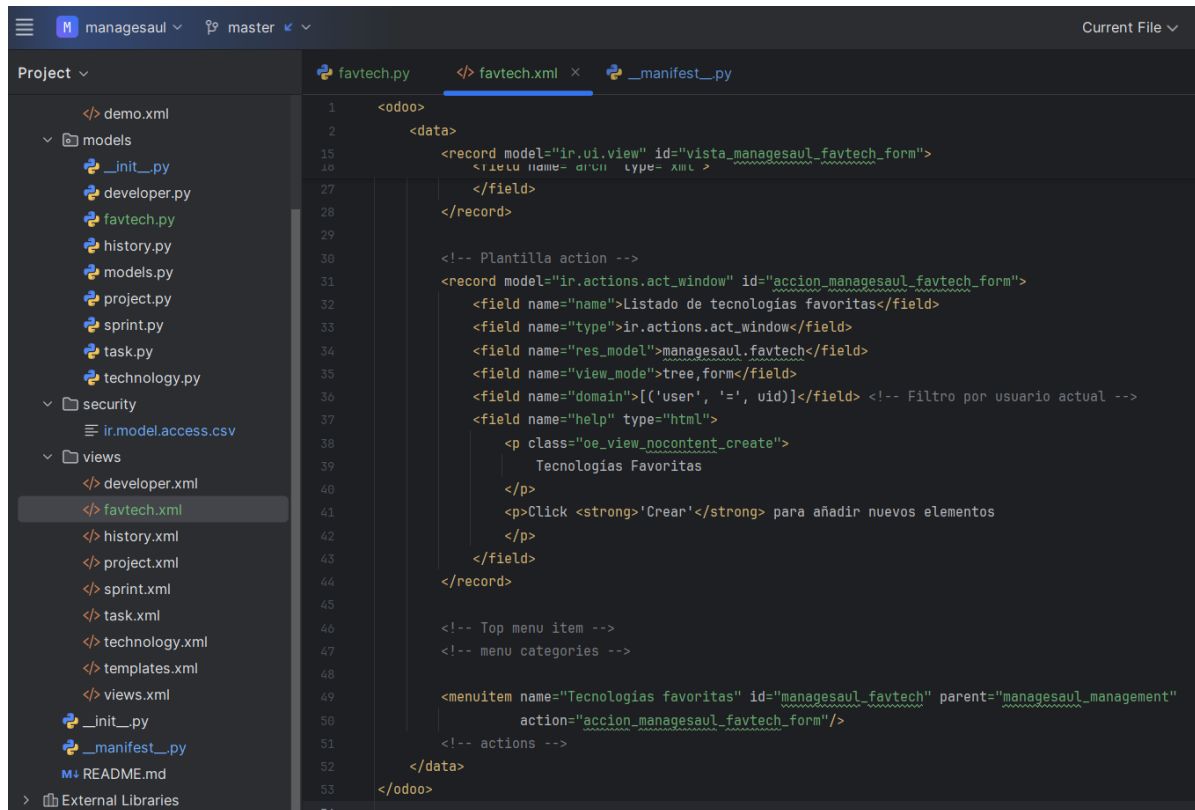
1 from odoo import models, fields
2
3 class favtech(models.Model):
4     _name = 'managesaul.favtech'
5     _description = 'managesaul.favtech'
6
7     favtech_id = fields.Many2one('managesaul.technology', string='Technology', required=True)
8     user = fields.Many2one(
9         'res.users',
10        string='User',
11        default=lambda self: self.env.user,
12        required=True
13    )
  
```



```

1 <odoo>
2     <data>
3         <record model="ir.ui.view" id="vista_managesaul_favtech_tree">
4             <field name="name">vista_managesaul_favtech_tree</field>
5             <field name="model">managesaul.favtech</field>
6             <field name="arch" type="xml">
7                 <tree>
8                     <field name="favtech_id"/>
9                     <field name="user"/>
10                </tree>
11            </field>
12        </record>
13
14        <record model="ir.ui.view" id="vista_managesaul_favtech_form">
15            <field name="name">vista_managesaul_favtech_form</field>
16            <field name="model">managesaul.favtech</field>
17            <field name="arch" type="xml">
18                <form string="formulario_favtech">
19                    <sheet>
20                        <group name="group_top">
21                            <field name="favtech_id"/>
22                            <field name="user"/>
23                        </group>
24                    </sheet>
25                </form>
26            </field>
27        </record>
28    </data>
29 </odoo>
  
```

Del xml cabe destacar la implementación del filtro de usuario. Gracias a este solo aparecerá la lista de tecnologías favoritas del usuario conectado en la sesión.



```

1 <odoo>
2 <data>
15 <record model="ir.ui.view" id="vista_managesaul_favtech_form">
16 <field name="arch" type="xml">
27 </field>
28 </record>
29
30 <!-- Plantilla action -->
31 <record model="ir.actions.act_window" id="accion_managesaul_favtech_form">
32 <field name="name">Listado de tecnologías favoritas</field>
33 <field name="type">ir.actions.act_window</field>
34 <field name="res_model">managesaul.favtech</field>
35 <field name="view_mode">tree,form</field>
36 <field name="domain">[('user', '=', uid)]</field> <!-- Filtro por usuario actual -->
37 <field name="help" type="html">
38 <p class="oe_view_nocontent_create">
39 Tecnologías Favoritas
40 </p>
41 <p>Click <strong>'Crear'</strong> para añadir nuevos elementos
42 </p>
43 </field>
44 </record>
45
46 <!-- Top menu item -->
47 <!-- menu categories -->
48
49 <menuitem name="Tecnologías favoritas" id="managesaul_favtech" parent="managesaul_management"
50 action="accion_managesaul_favtech_form"/>
51 <!-- actions -->
52 </data>
53 </odoo>

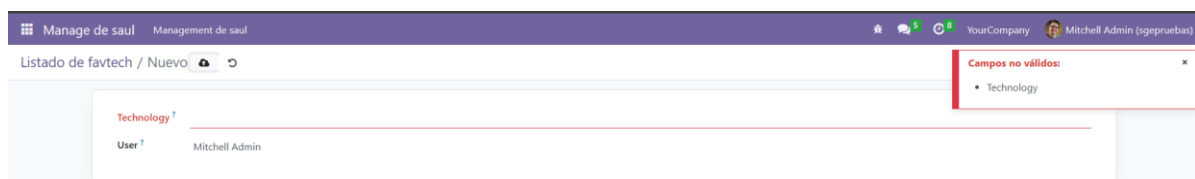
```

Respecto a los beneficios de la implementación cada usuario puede gestionar sus preferencias de manera independiente, sin modificar el modelo technology existente. La funcionalidad se puede extender fácilmente para incluir análisis sobre las tecnologías favoritas más comunes o integrarse con otros módulos.

7 Pruebas de funcionamiento.

7.1 Errores de usuario.

Cuando intentas crear una tecnología favorita sin poner esta sale un mensaje de error en rojo indicándote que no es válido el campo.



7.2 Edición de registros.

Comprobación que al editar un proyecto se actualizan los datos en la vista tree.

Manage de saul Management de saul	
Listado de project	
<div> <div>NUEVO</div> <div> <div>Buscar...</div> <div>Filtros Agrupar por Favoritos</div> </div> </div>	
1-1 / 1	
Nombre	Descripcion
<input type="checkbox"/> proyecto1	proyecto1

Después de ser editado:

Manage de saul Management de saul	
Listado de project	
<div> <div>NUEVO</div> <div> <div>Buscar...</div> <div>Filtros Agrupar por Favoritos</div> </div> </div>	
1-1 / 1	
Nombre	Descripcion
<input type="checkbox"/> proyectoEditado	proyecto1

7.3 Campos computados.

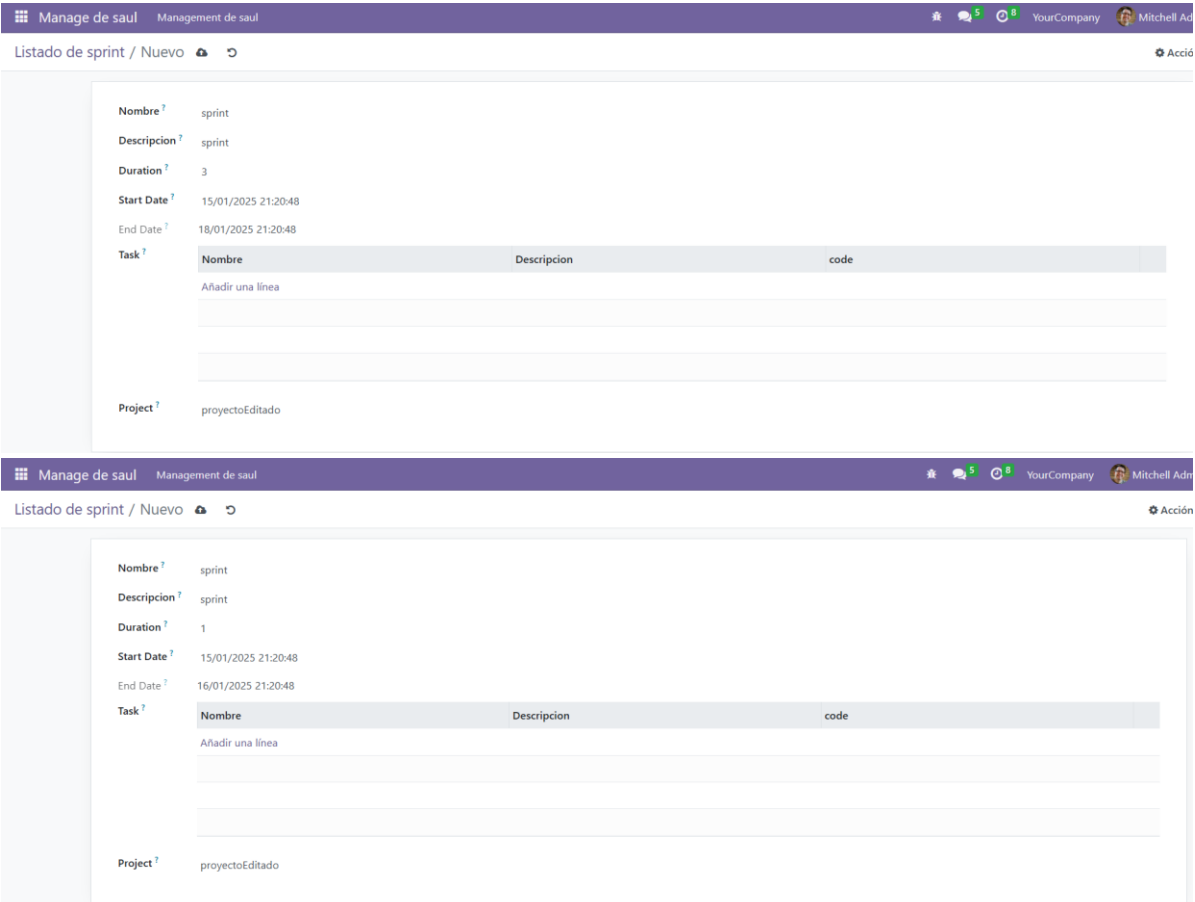
Al crear la tarea se crea solo el campo code y al asignar una historia el campo project se genera solo dependiendo del proyecto que este asociado a la historia seleccionada.

Manage de saul

Management de saul

</

En el modelo Sprint el campo end_date se calcula dependiendo de start_date y duration.



Manage de saul Management de saul YourCompany Mitchell Adn

Listado de sprint / Nuevo Acció

Nombre? sprint

Descripción? sprint

Duración? 3

Start Date? 15/01/2025 21:20:48

End Date? 18/01/2025 21:20:48

Task?

Nombre	Descripción	code
Añadir una línea		

Project? proyectoEditado

Manage de saul Management de saul YourCompany Mitchell Adn

Listado de sprint / Nuevo Acción

Nombre? sprint

Descripción? sprint

Duración? 1

Start Date? 15/01/2025 21:20:48

End Date? 16/01/2025 21:20:48

Task?

Nombre	Descripción	code
Añadir una línea		

Project? proyectoEditado

8 Conclusiones y posibles ampliaciones.

El proyecto ha sido creado para ofrecer una solución práctica y organizada en la gestión de proyectos, combinando herramientas útiles para planificar tareas, registrar cambios importantes y permitir a los usuarios personalizar su experiencia seleccionando sus tecnologías favoritas.

A través de sus funcionalidades, permite llevar un control claro del trabajo en equipo, facilitando el seguimiento de tareas, el registro de avances y la organización de los ciclos de trabajo. Además, con la opción de elegir tecnologías favoritas, cada usuario puede personalizar su experiencia y acceder rápidamente a lo que más necesita.

Como posible ampliación se podría implementar un historial de cambios en las tareas, permitiendo llevar un registro de los cambios realizados en estas. Cada vez que se hiciera alguna modificación en algún campo de una tarea se guardaría el registro en un modelo creado para ello junto con el usuario que realizó la modificación.

9 Github.

<https://github.com/saulmantecon/managesaul.git>

10 Bibliografía.

Temario de SGE.

ChatGPT.

<https://www.atlassian.com/es/agile/scrum>