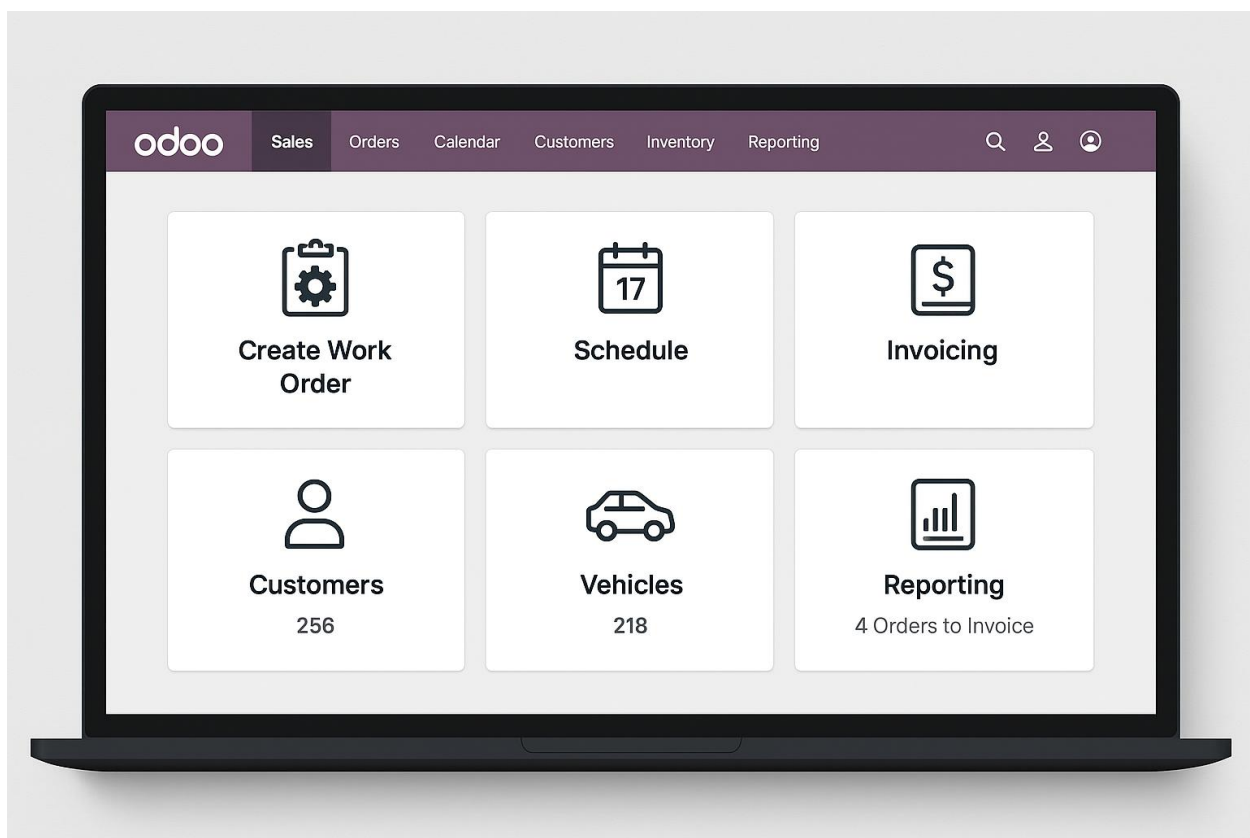


Projecte Final de Cicle

DIGITALITZACIO D'UN TALLER MECANIC



Alumne:	Jose Miguel Cendan Cabanilles
DNI:	20039854T
Tutor Individual:	Alicia González Canet
Tutor del grup:	Jose Alfredo Murcia Andres

Dades del Projecte

Dades de l'alumne

Nom i cognoms	Jose Miguel Cendan Cabanilles
NIF/NIE	20039854T
Curs i CF	2n DAM

Dades del projecte

Títol del projecte	DIGITALITZACIO D'UN TALLER MECANIC
Nom del tutor individual	Alicia González Canet
Nom del tutor del grup	Jose Alfredo Murcia Andres
Resum	<i>El projecte consisteix en el desenvolupament d'un mòdul personalitzat per a Odoo, destinat a gestionar un taller mecànic. El sistema permet registrar vehicles, associar-los a clients, crear pressupostos de reparacions i generar factures automàticament. També integra la facturació amb els mòduls oficials d'Odoo i ofereix una vista centralitzada de les operacions del taller. El desenvolupament s'ha realitzat íntegrament en Python i XML, seguint l'arquitectura de mòduls d'Odoo.</i>
Abstract	<i>The project consists of developing a custom module for Odoo aimed at managing a mechanical workshop. The system allows registering vehicles, linking them to customers, creating repair quotes, and generating invoices automatically. It integrates with Odoo's official accounting module and provides a centralized view of workshop activity. The project is entirely developed using Python and XML within Odoo's modular architecture.</i>
Mòduls implicats	<ul style="list-style-type: none">• Programació• Bases de dades• Sistemes de gestió empresarial• Desenvolupament d'interfícies
Data de presentació	03 de juny de 2025

Índex

Dades del Projecte.....	1
Dades de l'alumne.....	1
Dades del projecte.....	1
1. Introducció/Marc del projecte	3
1.1. Descripció del projecte.....	3
1.2. Objectius	3
1.3. Tipus de projecte.....	3
1.4. Orientacions per al desenvolupament i recursos	3
2. Anàlisi de Requeriments.....	7
2.1. Estudi de l'estat de l'art/Estudi de mercat	7
2.2. Definició de requeriments	7
3. Metodologia de Desenvolupament.....	11
3.1. Descripció de la metodologia àgil.....	11
3.2. Backlog del producte	11
3.3. Planificació temporal.....	14
4. Desenvolupament iteratiu per sprints.....	15
El projecte s'ha organitzat en sprints de 1-2 setmanes, seguint una planificació adaptativa basada en el backlog prioritzat. A continuació es detalla l'execució de cada sprint, els objectius assolits i els lliuraments clau:	15
4.1. Sprint 1: (5/05 - 12/05): Model Bàsic + Vistes Inicials.....	15
4.2.Sprint2(13/05-24/05):Funcionalitat de pressupostos i facturació.....	17
4.3. Sprint 3: (25/05 - 30/05): Ajustos finals i preparació per a producció	18
5. Conclusions i Treballs Futurs	19
5.1. Resum dels resultats	19
El projecte "Digitalització d'un Taller Mecànic" ha permès desenvolupar un mòdul funcional dins de l'ERP Odoo que centralitza la gestió de clients, vehicles, reparacions i facturació. Els principals assoliments inclouen:	19
5.2. Possibles millores	20

1. Introducció/Marc del projecte

1.1. Descripció del projecte

El projecte consisteix en el desenvolupament d'un mòdul personalitzat per al sistema ERP Odoo que permeti gestionar el dia a dia d'un taller mecànic d'una persona. L'aplicació cobreix funcionalitats bàsiques com el registre de clients i vehicles, la generació de pressupostos per a reparacions, la conversió d'aquests en factures, i el seguiment administratiu. El mòdul integrar-se amb els mòduls oficials de vendes i comptabilitat d'Odoo.

1.2. Objectius

- Desenvolupar un mòdul funcional dins de l'ecosistema Odoo.
- Gestionar clients i vehicles de forma vinculada.
- Crear pressupostos de reparacions amb dades detallades.
- Generar factures automàticament des dels pressupostos.
- Aprendre a personalitzar i ampliar l'ERP Odoo.
- Aplicar metodologies àgils durant el desenvolupament.
- Documentar el procés de manera clara i estructurada.

1.3. Tipus de projecte

- **Intern:** El projecte està pensat per ser utilitzat dins d'un entorn real (un taller mecànic) com a eina de gestió pròpia.

1.4. Orientacions per al desenvolupament i recursos

El projecte s'ha desenvolupat com un mòdul personalitzat dins de l'entorn Odoo, una plataforma ERP de codi obert molt potent i flexible, utilitzada àmpliament per empreses per gestionar processos interns (vendes, compres, facturació, inventari, etc.).

S'ha escollit Odoo per la seua arquitectura modular, la possibilitat d'adaptar-lo a diferents sectors i la facilitat per integrar funcionalitats noves mitjançant mòduls propis.



➤ **Programació**

S'ha utilitzat Python com a llenguatge principal per desenvolupar la lògica de negoci del mòdul. Aquest llenguatge és nadiu en Odoo i permet crear models, accions i automatismes mitjançant decoradors (@api) i herències del framework. La seua simplicitat i potència fan que sigui ideal per desenvolupar funcionalitats ràpidament dins d'un entorn ERP.



➤ **Interfície d'usuari**

La definició de vistes s'ha fet mitjançant XML, tal com Odoo exigeix. Això inclou formularis, llistats, filtres de cerca, menús i accions. Aquesta estructura facilita la personalització del comportament de les pantalles per adaptar-les a les necessitats específiques del taller.



➤ **Base de dades**

Odoo funciona sobre PostgreSQL, que és el sistema de base de dades relacional utilitzat. La interacció amb les dades es realitza mitjançant l'ORM d'Odoo, que permet manipular registres com objectes Python, simplificant molt les consultes i l'estructura del codi.



➤ **Recursos de suport i documentació**

Per dur a terme el desenvolupament s'han utilitzat diverses fonts de suport:

- Documentació oficial d'Odoo, per comprendre l'estructura dels mòduls i el funcionament intern de la plataforma.



- Fòrums de la comunitat (com Stack Overflow i Odoo Community Association) per resoldre dubtes tècnics concrets i veure casos pràctics.



- Vídeos de YouTube, especialment tutorials tècnics sobre desenvolupament de mòduls, configuració d'Odoo i integracions.



- ChatGPT, utilitzat com a assistent per resoldre dubtes de programació, revisar codi, generar exemples i redactar parts del projecte.



- Visual Studio Code (VS Code), l'editor de codi principal per al desenvolupament. Gràcies a la seva extensibilitat i suport per a Python, XML i PostgreSQL, facilita l'escriptura i depuració del codi. A més, incorpora extensions específiques per treballar amb Odoo i permet integrar terminal i control de versions Git.



- Docker, que s'ha emprat per containeritzar l'entorn de desenvolupament i desplegament. Amb Docker, es garanteix que el mòdul funcioni de manera consistent en diferents entorns, simplificant la instal·lació i configuració de PostgreSQL, Odoo i les dependències del projecte. Això també facilita la col·laboració entre desenvolupadors i la gestió de versions del projecte.



- Mermaid, eina per a la creació de diagrames en format text. Permet generar diagrames UML com ara classes, seqüències i fluxos de treball de forma senzilla i integrada en documentació Markdown. Gràcies a la seva sintaxi clara i suport en plataformes com GitHub, facilita la visualització de l'arquitectura del sistema i el comportament del codi, millorant la claredat de la documentació tècnica del projecte.



2. Anàlisi de Requeriments

2.1. Estudi de l'estat de l'art/Estudi de mercat

Existeixen diversos sistemes de gestió utilitzats habitualment en tallers mecànics, alguns desenvolupats a mida i altres basats en fulls de càlcul com Microsoft Excel o LibreOffice Calc. Aquestes solucions, tot i ser accessibles i conegudes, presenten limitacions pel que fa a l'automatització, la integració de processos i la traçabilitat de la informació.

Aquest projecte proposa una alternativa professional mitjançant Odoo, una plataforma ERP de codi obert que permet una integració completa entre la gestió de clients, vehicles, pressupostos i facturació, aportant més control i escalabilitat.

2.2. Definició de requeriments

- Requeriments funcionals i no funcionals.

Requeriments funcionals:

Perquè l'aplicació complisca amb l'objectiu planejat, serà necessari que dispose de les següents funcionalitats:

- El sistema ha de permetre crear noves ordres de reparació.
- L'usuari pot editar els detalls d'una orde existent.
- El sistema ha de permetre eliminar ordres de reparació sol si no estan en procés.
- L'usuari ha de poder buscar ordres per client, vehicle o data.
- S'ha de permetre l'assignació de tècnics a ordres específiques.
- El sistema ha de registrar l'historial de canvis realitzats en cada orde.
- Els usuaris han de poder exportar dades d'ordres en format CSV o PDF.
- El sistema ha de validar les dades obligatòries abans de guardar una orde.
- El sistema ha de gestionar els permisos d'accés segons rols (mecànic, administrador, client).
- El sistema enviarà alertes per correu electrònic per a ordres pendents d'aprovació.
- El sistema ha de mantindre un registre de clients i vehicles vinculats.
- S'ha de permetre la creació i gestió de factures associades a ordres completades.

Requeriments no funcionals:

Amb la finalitat que l'aplicació proporcione una bona experiència per als usuaris, i ofereixi facilitat d'ús, també hauríem de dotar-la d'estes característiques.

- El sistema ha de respondre a les sol·licituds en menys de 2 segons.

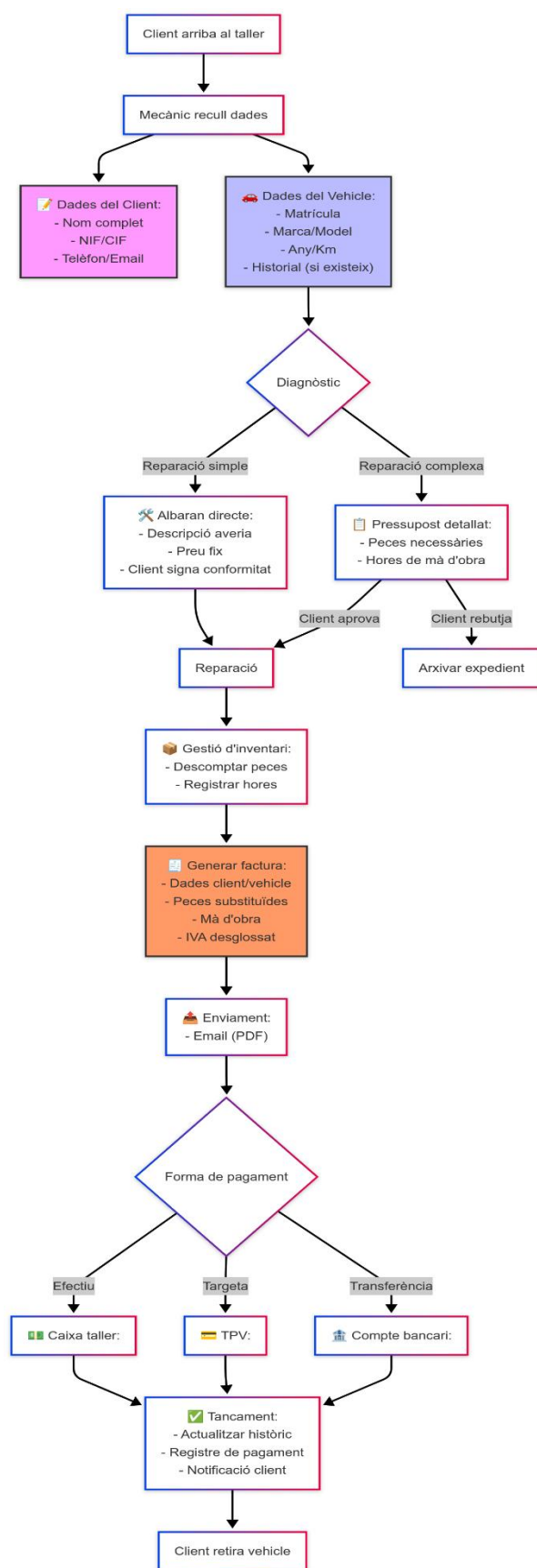
- La disponibilitat del sistema ha de ser del 99,9% anual.
- El sistema ha de ser compatible amb els navegadors Chrome, Firefox i Edge.
- El temps màxim d'inactivitat planificada no ha de superar les 4 hores mensuals.
- El sistema ha de complir amb la normativa GDPR per a protecció de dades.
- La interfície ha de ser intuïtiva i fàcil d'usar per a usuaris amb coneixements bàsics.
- El sistema ha de ser escalable per a futures ampliacions.
- La documentació tècnica i d'usuari ha d'estar disponible i actualitzada.
- El sistema ha de permetre la integració amb sistemes externs mitjançant APIs.
- Les dades han d'emmagatzemar-se en un format que permeti exportació i importació senzilla.
- El sistema ha de manejar errors de manera robusta i mostrar missatges clars a l'usuari.
- Casos d'ús i històries d'usuari.

Casos d'Ús Principals

1. CU-01: Registrar entrada de vehicle
 - Actor: Client
 - Descripció: Client proporciona dades personals i del vehicle al mecànic.
 - Precondició: Vehicle present al taller.
2. CU-02: Realitzar diagnòstic
 - Actor: Mecànic
 - Descripció: Identifica l'averia i decideix si requereix pressupost.
3. CU-03: Aprovar reparació
 - Actor: Client
 - Descripció: Client signa albarans o aprova pressupost per email.
4. CU-04: Gestió de pagament
 - Actor: Client/Caixer
 - Descripció: Client tria mètode de pagament (efectiu, targeta, etc.).

Històries d'Usuari Relacionades

ID	Història	Prioritat
01	"Com a client, vull rebre un pressupost detallat abans d'aprovar reparacions costoses per prendre decisions."	Alta
02	"Com a mecànic, necessita accedir a l'historial de reparacions del vehicle per agilitzar el diagnòstic."	Crítica
03	"Com a administrador, vull que el sistema generi factures automàticament amb desglossament d'IVA per complir amb la llei."	Mitja



Taula de Responsables Detallada

Pas	Responsable	Accions Clau	Document/Registre Generat	Col·laboradors
1. Recepció	Mecànic	- Recollir dades client (DNI, contacte) - Registrar matrícula i km vehicle	Fitxa tècnica inicial	Client
2. Diagnòstic	Mecànic	- Identificar averia - Consultar historial - Decidir si fa albaran o pressupost	Informe diagnòstic (escrit/digital)	Sistema (base de dades)
3. Aprovació	Client	- Signar albaran- Respondre email per pressupost	Albaran signat o gravació digital d'aprovació	Mecànic
4. Reparació	Mecànic	- Utilitzar peces - Registrar hores treballades - Fer proves post-reparació	Ordre de treball completada	Mecànic
5. Facturació	Sistema	- Generar PDF amb IVA desglossat - Vincular garanties	Factura numèrica (format legal)	Mecànic
6. Pagament	Client/Caixa	- Triar mètode pagament - Registrar entrada de diners (TPV/efectiu)	Rebut de pagament (paper/digital)	Sistema bancari
7. Lliurament	Mecànic	- Entregar claus - Explicar reparació feta - Lliurar factura original	Check-list de lliurament signat	Client

3. Metodologia de Desenvolupament

3.1. Descripció de la metodologia àgil

Encara que Scrum està dissenyat per a equips, he aplicat una versió simplificada per a gestionar el projecte de manera àgil i eficient:

- **Sprints:** Períodes de 1-2 setmanes amb objectius concrets.
- **Backlog:** Llista de tasques prioritzades (GitHub Projects).
- **Reunions amb mi mateix:**
 - **Autorevisió diària** (10 min): Què vaig fer ahir? Què faré hui? Quins obstacles tinc?
 - **Sprint Planning:** Seleccione tasques del backlog per al següent sprint.
 - **Sprint Retrospective:** Avalue què millorar en el pròxim sprint.

3.2. Backlog del producte

Les funcionalitats es prioritzen segons complexitat i temps disponible:

Prioritat	Funcionalitat	Detalls Tècnics	Temps Estimats
Alta	Model principal personalitzat	- Crear un model nou (vehicle, taller). - Camps: nom, propietari, tècnic....	1 setmana
Alta	Vistes personalitzades (XML)	- Formularis i llistes per al model principal. - Filtres i cerques ràpides.	1 setmana
Mitjana	Informes/QWeb	- Generar PDFs amb dades del model (ex: llistat de tasques).	1 setmana
Baixa	Integració amb altres mòduls	- Enllaçar amb <code>hr.sale</code> o <code>hr.employee</code> si s'escau.	0.5 setmanes

1. Model bàsic:

```
class Factura(models.Model):
    _inherit = 'account.move'

    kilometros = fields.Integer(string="Kilómetros vehículo")
    vehiculo_ids = fields.Many2many(
        'taller.vehiculo',
        string="Vehículos relacionados",
        relation='factura_vehiculo_rel',
        column1='factura_id',
        column2='vehiculo_id'
    )

    @api.depends('partner_id')
    def _compute_vehiculo_ids(self):
        for move in self:
            if move.partner_id:
                move.vehiculo_ids = self.env['taller.vehiculo'].search([
                    ('cliente_ids', 'in', move.partner_id.ids)
                ])
            else:
                move.vehiculo_ids = False
```

2. Vista XML (formulari):

```
<odoo>
  <record id="view_order_form_inherit_vehiculos" model="ir.ui.view">
    <field name="name">sale.order.form.inherit.vehiculos</field>
    <field name="model">sale.order</field>
    <field name="inherit_id" ref="sale.view_order_form"/>
    <field name="arch" type="xml">
      <xpath expr="//field[@name='partner_id']" position="after">
        <field name="vehiculo_ids"
              widget="many2many_tags"
              domain="['('cliente_id', 'in', [partner_id])']"
              options='{\'no_create\': True}'
              string="Vehículos relacionados"/>
        <field name="kilometros" placeholder="Introduce los kilómetros actuales"/>
      </xpath>
    </field>
  </record>
</odoo>
```

3. Informes



SG Tallers
Av.Valencia,13B 03770-El Verger Alacant España

Ines Garcia
Alfaro
46702 gandia (València (Valencia))
España
NIF:

Factura INV/2025/00001

Fecha de factura:
26/05/2025

Fecha de vencimiento:
26/05/2025

Vehículos relacionados:

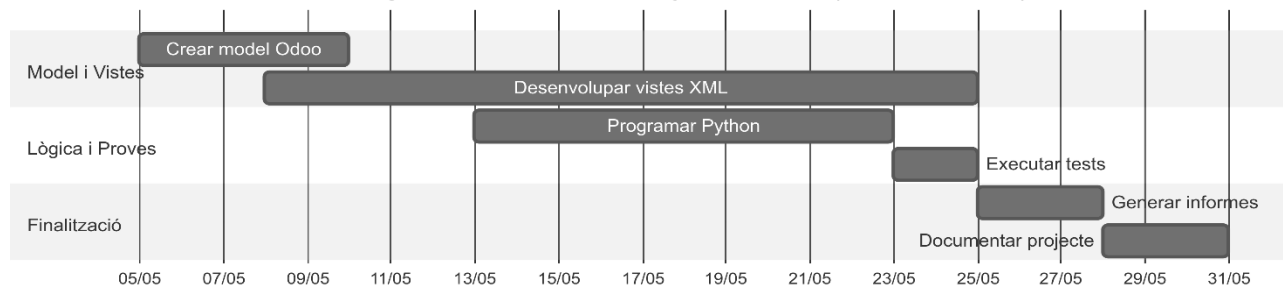
- Matrícula: 0327FBT, Marca: FORD, Modelo: FOCUS

Kilómetros del Vehículo: 555

Descripción	Cantidad	Precio unitario	Impuestos	Importe
Frenos de disco	1,00	1,00	IVA 21% (Bienes)	1,00 €
Base imponible				1,00 €
IVA 21%				0,21 €
Total				1,21 €

Por favor utilice la siguiente referencia al realizar su pago: **INV/2025/00001**

Diagrama de Gantt - Projecte Taller (5/05 al 30/05)



3.3. Planificació temporal

Cronograma compacte:

Fase	Dates	Dies	Solapament amb
Model	5-9/05	5	Vistes (día 8)
Vistes	8-27/05	17	Python (13/05)
Python	13-24/05	10	Tests (23/05)
Tests	23-24/05	2	Informes (25/05)
Informes	25-27/05	3	Documentació (28/05)
Documentació	28-30/05	3	-

4. Desenvolupament iteratiu per sprints

El projecte s'ha organitzat en sprints de 1-2 setmanes, seguint una planificació adaptativa basada en el backlog prioritzat. A continuació es detalla l'execució de cada sprint, els objectius assolits i els lliuraments clau:

4.1. Sprint 1: (5/05 - 12/05): Model Bàsic + Vistes Inicials

Objectiu: Establir l'estructura central del mòdul Odoo per a la gestió del taller mecànic.

- **Disseny de la solució**
 - Creació del model principal taller amb camps bàsics (client, vehicle, data, estat, descripció de l'averia).
 - Definició de les relacions amb els models existents d'Odoo (res.partner per a clients).
 - Desenvolupament de vistes XML inicials: formulari per a registrar reparacions i llista per a visualitzar-les.
 - Configuració de menús i accions per a accedir al mòdul des de la interfície d'Odoo.
- **Implementació**
 - Model Vehiculo

```
class Vehiculo(models.Model):  
    _name = 'taller.vehiculo'  
    _description = 'Vehículo'  
  
    matricula = fields.Char(string="Matrícula", required=True)  
    numero_chasis = fields.Char(string="Número de chasis", required=True, index=True)  
    marca = fields.Char()  
    modelo = fields.Char()  
    fecha_matriculacion = fields.Date(string="Fecha de matriculación")  
    cliente_id = fields.Many2one('res.partner', string="Propietario", required=True)
```

- Relació

```
class Partner(models.Model):  
    _inherit = 'res.partner'  
  
    vehiculo_ids = fields.One2many('taller.vehiculo', 'cliente_id', string="Vehículos del cliente", readonly=True)
```


- Vistes XML

```
<odoo>
<record id="view_taller_orden_form" model="ir.ui.view">
  <field name="name">taller.orden.form</field>
  <field name="model">taller.orden</field>
  <field name="arch" type="xml">
    <form string="Orden de Reparación">
      <header>
        <button name="crear_factura"
          type="object"
          string="Crear Factura"
          class="btn-primary"
          attrs="{ 'invisible': [ '|', ('factura_id', '!=', False), ('estado', '!=', 'completado') ] }" />

        <button name="action_open_factura"
          type="object"
          string="Ver Factura"
          class="btn-secondary"
          attrs="{ 'invisible': [ ('factura_id', '=', False) ] }" />
      </header>

      <sheet>
        <field name="factura_id" invisible="1"/>
        <group>
          <field name="vehiculo_id"/>
          <field name="tecnico_id"/>
          <field name="fecha_inicio"/>
          <field name="fecha_fin"/>
          <field name="estado"/>
          <field name="kilometros"/>
        </group>
        <notebook>
          <page string="Repuestos">
            <field name="repuesto_ids" context="{ 'default_orden_id': active_id }" create="true" edit="true">
              <tree editable="bottom">
                <field name="producto_id"/>
                <field name="cantidad"/>
              </tree>
            </field>
          </page>
        </notebook>
      </sheet>
    </form>
  </field>
</record>
```

- Accions i menús

```
<record id="action_clientes_taller" model="ir.actions.act_window">
  <field name="name">Clientes del Taller</field>
  <field name="res_model">res.partner</field>
  <field name="view_mode">tree,form</field>
  <field name="domain">[('vehiculo_ids', '!=', False)]</field>
  <field name="context">{'search_default_customer': 1}</field>
</record>

<menuitem id="menu_taller_root" name="Taller" sequence="10"/>
<menuitem id="menu_taller_clientes_taller_lista" name="Lista de Clientes" parent="menu_taller_root" action="action_clientes_taller" sequence="10"/>
```

- Proves



The screenshot shows a web application with a navigation bar at the top containing links: 'Lista de Clientes', 'Vehículos', 'Albaranes', 'Facturas', 'Presupuestos', and 'Productos'. Below the navigation bar is a header area with a 'Nuevo' button and two icons. The main content area is a form for creating a new vehicle record. The form has a light blue background and contains the following fields: 'Matrícula?' with a help icon and an example 'Ejemplo: M123456, AB1234CD, 1234ABC, C1234ABC, R1234ABC'; 'Fecha de matriculación?'; 'Número de chasis?'; 'Marca?'; 'Modelo?'; and 'Propietario?'. A small circle is visible at the bottom left of the form area.

4.2.Sprint2(13/05-24/05):Funcionalitat de pressupostos i facturació

Objectiu: Implementar la generació de pressuposts i la seva conversió automàtica a factures.

- Disseny de la solució

- Ampliació del model factura.py amb camps per a introduir dades (vehicle,Kilometres).
- Desenvolupament d'un informe QWeb per a generar factures en PDF.
- Automatització de la creació de factures mitjançant l'API d'Odoo (account.move).

- Implementació

- Model Factura.py
- QWeb.
- Automatitzacio factures.

```
class Factura(models.Model):  
    _inherit = 'account.move'  
  
    kilometros = fields.Integer(string="Kilómetros vehículo")  
    vehiculo_ids = fields.Many2many(  
        'taller.vehiculo',  
        string="Vehículos relacionados",  
        relation='factura_vehiculo_rel',  
        column1='factura_id',  
        column2='vehiculo_id'  
    )
```

- **Proves**

Vehículos relacionados:

- Matrícula: 0327FBT, Marca: FORD, Modelo: FOCUS

Kilómetros del Vehículo: 555

✉ Enviar Factura

☒ Imprimir? Vista previa como un PDF

☐ Enviar por correo postal?

☒ Correo electrónico?

Destinatarios? Seguidores del documento y

Ines Garcia ✕

Asunto? SG Tallers Factura (Ref INV/2025/00008)

Estimado/a Ines Garcia,

Aquí está su factura **INV/2025/00008** (con la referencia: Orden de Reparación #2) por un importe total de **48,40 €** de parte de SG Tallers. Por favor, revise la forma de pago.

Utilice la siguiente comunicación para el pago: **INV/2025/00008**.

No dude en contactarnos si tiene alguna pregunta.

System



Factura_INV_2025_00008.pdf
PDF



Usar plantilla?

Factura: enviando

ARCHIVOS ADJUNTOS

4.3. Sprint 3: (25/05 - 30/05): Ajustos finals i preparació per a producció

Objectiu: Polir funcionalitats, optimitzar el rendiment i preparar el lliurament final.

- **Disseny de la solució**

- Millores a la interfície d'usuari: afegir filtres ràpids per estat i client.
- Implementació de permisos basats en rols (mecànic, administrador).
- Proves de càrrega amb dades de prova

- **Implementació**

- Busques.

```
<record id="view_taller_vehiculo_search" model="ir.ui.view">
  <field name="name">taller.vehiculo.search</field>
  <field name="model">taller.vehiculo</field>
  <field name="arch" type="xml">
    <search string="Buscar Vehículo">
      <field name="matricula"/>
      <field name="numero_chasis"/>
      <field name="marca"/>
      <field name="modelo"/>
      <field name="fecha_matriculacion"/>
    </search>
  </field>
</record>
```

- **Proves**



5. Conclusiones i Treballs Futurs

5.1. Resum dels resultats

El projecte "Digitalització d'un Taller Mecànic" ha permès desenvolupar un mòdul funcional dins de l'ERP Odoo que centralitza la gestió de clients, vehicles, reparacions i facturació. Els principals assoliments inclouen:

- **Implementació tècnica satisfactòria:**
 - Creació d'un model personalitzat (vehicle.repair) amb Python i vistes XML per a una interfície intuïtiva.
 - Automatització de processos clau: generació de pressupostos, conversió a factures i vinculació amb els mòduls de comptabilitat d'Odoo.
 - Integració amb els mòduls estàndard d'Odoo (Sales, Employee, Account, Respartner).
- **Aplicació pràctica:**
 - El sistema redueix errors manuals en la facturació i millora la traçabilitat de les reparacions.
 - Els usuaris poden accedir a dades centralitzades i generar informes en PDF amb facilitat.
- **Aprenentatge adquirit:**
 - Domini de l'arquitectura modular d'Odoo i el seu ORM.
 - Ús de metodologies àgils (Scrum adaptat) per a gestionar tasques en solitari.

Dificultats superades:

- Complexitat en la vinculació de models personalitzats amb els mòduls oficials d'Odoo (ex: facturació).
- Optimització del rendiment en consultes amb grans volums de dades.

5.2. Possibles millores

El projecte obre la porta a futures ampliacions, entre les quals es destaquen:

- Funcionalitats avançades:
 - Notificacions automàtiques: Enviar WhatsApp o correus electrònics als clients quan el seu vehicle estigui llest.
 - Copia de seguridad automàtica.
 - Auditoría de cambios críticos

Reflexió final:

Aquest projecte no només ha complert els objectius inicials sinó que també ha posat les bases per a escalar la solució. La flexibilitat d'Odoo permet adaptar-se a necessitats futures, i l'experiència adquirida serà útil per a desenvolupaments similars.