

ODOO

# GESTIÓ D'EQUIPS I EMPLEATS



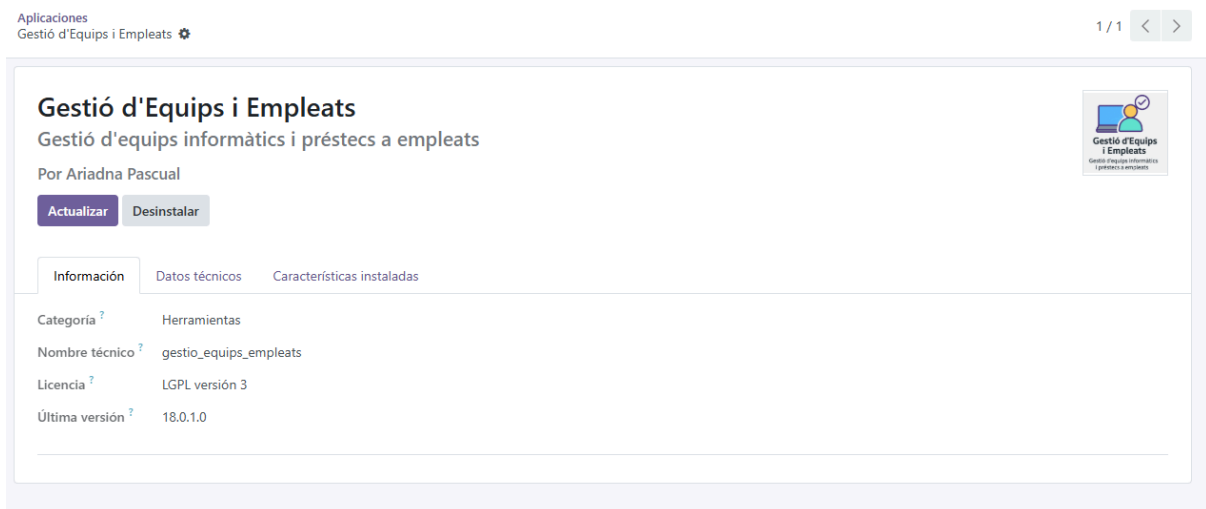
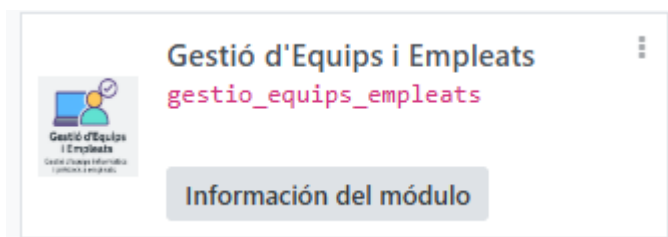
## ÍNDEX

1. Introducció
2. Instal·lació i creació de la base de dades
3. Estructura del mòdul
  - 3.1. Manifest i arxius d'inicialització
  - 3.2. Models i relacions
4. Desenvolupament del mòdul
  - 4.1. Model empleado
  - 4.2. Model equipo
  - 4.3. Model préstamo
  - 4.4. Validacions i constraints
5. Menús i vistes XML
6. Impressió de l'informe PDF (QWeb)
7. Enviament de l'informe per correu
8. Assistència IA integrada amb OpenAI
9. Dades de demostració
10. Permisos d'accés i rols d'usuari
11. Assignació de Grups a Usuaris
12. Traducció amb Poedit
13. Firma digital del Treballador
14. Problemes trobats i Millores
15. Conclusions

## 1. Introducció

Aquest projecte consisteix en el desenvolupament complet d'un mòdul personalitzat per a Odoo, anomenat "Gestió d'Equips i Empleats". L'objectiu és gestionar el préstec d'equips informàtics a empleats, amb funcionalitats de registre, control, validació, generació d'informes en pdf, enviament per correu, assistència intel·ligent mitjançant una IA ChatGPT

El projecte s'ha desenvolupat amb una estructura modular clara, implementant bones pràctiques de desenvolupament i seguint l'arquitectura d'Odoo.



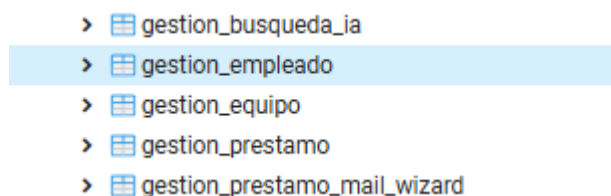
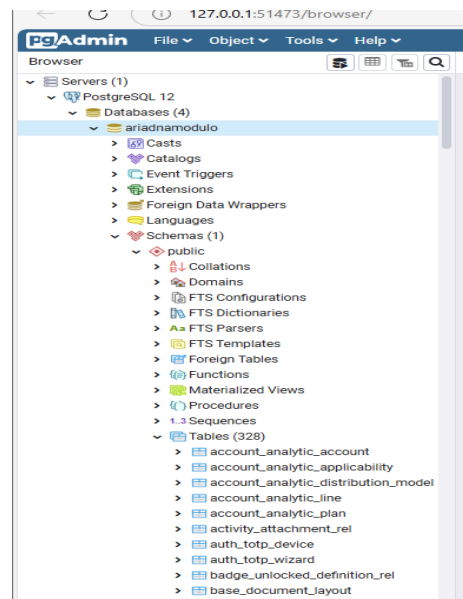
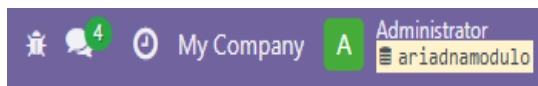
## 2. Instal·lació i creació de la base de dades i mòdul

Inicialment, es va crear una base de dades anomenada **ariadnapascual**, però per problemes tècnics va ser substituïda per una nova base de dades anomenada **ariadnamodulo**, on s'ha desenvolupat completament el projecte. Aquesta base de dades està activa amb Odoo 18 Community Edition.

Aquesta base de dades es troba dins del servidor PostgreSQL 12, que és compatible amb Odoo 18.0. A ariadnamodulo s'han anat guardant totes les dades relacionades amb el mòdul **gestio\_equipos\_empleats**, incloent els models, vistes, dades de demostració, grups d'usuaris, informes QWeb i interaccions amb l'assistent IA.

Amb PostgreSQL i pgAdmin he pogut:

- Verificar que les taules **gestion\_empleado**, **gestion\_equipo** i **gestion\_prestamo** es generaven correctament.
- Confirmar que les claus foranes Many2one i One2many estaven ben creades.



Es van habilitar les opcions de desenvolupador per permetre la personalització completa del sistema, així com l'accés a les funcionalitats tècniques.

Per iniciar el desenvolupament del mòdul **gestio\_equipos\_empleats**, vaig utilitzar la comanda **scaffold** que proporciona Odoo 18.0. Aquesta eina permet generar automàticament l'estructura bàsica i necessària d'un mòdul personalitzat, facilitant l'organització inicial del projecte.

#### **Entorn de treball:**

- **Sistema:** Windows
- **Versió d'Odoo:** 18.0 Community
- **Lloc des d'on s'executa la comanda:**  
C:\Program Files\Odoo 18.0.20250127\python
- **Ruta de l'arxiu **odoo-bin**:**  
"C:\Program Files\Odoo 18.0.20250127\server\odoo-bin"
- **Ruta de desplegament dels mòduls (addons):**  
"C:\Program Files\Odoo 18.0.20250127\server\odoo\addons"

#### **Comanda executada:**

```
python.exe "C:\Program Files\Odoo 18.0.20250127\server\odoo-bin" scaffold  
gestio_equipos_empleats "C:\Program Files\Odoo 18.0.20250127\server\odoo\addons"
```

Després d'executar aquesta comanda, es va crear automàticament una carpeta amb el nom del mòdul **gestio\_equipos\_empleats**, que contenia l'estructura mínima per començar el desenvolupament:

- `__init__.py` i `__manifest__.py`

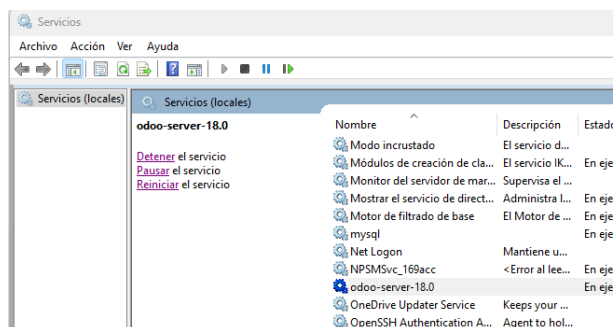
## Manifest

```
{
  'name': "Gestió d'Equips i Empleats",
  'version': '1.0',
  'summary': "Gestió d'equips informàtics i préstecs a empleats",
  'description': "Gestió d'equips, préstecs, empleats i enviament d'informes.",
  'author': "Ariadna Pascual",
  'category': 'Tools',
  'license': 'LGPL-3',
  'depends': ['base', 'mail'],
  'data': [
    'security/security.xml',
    'security/ir.model.access.csv',
    'wizard/prestamo_mail_wizard_view.xml',
    'wizard/prestamo_report_template.xml',
    'wizard/prestamo_report.xml',
    'views/menu_views.xml',
    'views/equipo_views.xml',
    'views/prestamo_views.xml',
    'views/empleado_views.xml',
    'views/busqueda_ia_views.xml',
  ],
  'demo': [
    'demo/demo.xml',
  ],
  'installable': True,
  'application': True,
  'auto_install': False,
}
```

- Carpetes: models/, views/, security/, static/, data/, wizard/, etc.

A partir d'aquesta base, vaig començar a desenvolupar la lògica de negoci, els models de dades (empleats, equips, préstecs), i les funcionalitats visuals del mòdul.











Per anar actualitzant el mòdul:



### 3. Estructura del Codi

El mòdul està organitzat segons la convencions d'Odoo:

- `__manifest__.py`: Descripció del mòdul, dependències i arxius carregats.
- `__init__.py`: Importació dels models.
- `models/`: Conté les classes `gestion.equipo`, `gestion.empleado`, `gestion.prestamo`, i altres.
- `views/`: Vistes XML per formularis, llistes, menús i accions.
- `wizard/`: Conté l'assistent per enviar informes per correu (`prestamo_mail_wizard.py`) i la seva vista XML.
- `security/`: Arxius `security.xml` per definir grups i permisos, i `ir.model.access.csv` per controlar accés als models.
- `data/demo.xml`: Arxiu de dades de demostració per equips i un usuari.
- `static/description/icon.png`: Icona del mòdul utilitzada en informes.

	<code>__pycache__</code>	15/04/2025 11:58	Carpeta de archivos	
	<code>demo</code>	15/04/2025 8:53	Carpeta de archivos	
	<code>i18n</code>	15/04/2025 15:32	Carpeta de archivos	
	<code>models</code>	15/04/2025 10:05	Carpeta de archivos	
	<code>security</code>	15/04/2025 10:28	Carpeta de archivos	
	<code>static</code>	15/04/2025 7:30	Carpeta de archivos	
	<code>views</code>	15/04/2025 10:07	Carpeta de archivos	
	<code>wizard</code>	15/04/2025 11:58	Carpeta de archivos	
	<code>__init__</code>	15/04/2025 11:34	Archivo de origen ...	1 KB
	<code>__manifest__</code>	15/04/2025 12:05	Archivo de origen ...	1 KB

\* el `i18n` es una ampliació del mòdul

## 4. Models, Funcionalitats i Menús/Submenús

### Menús i Interfície del Mòdul

L'estructura de navegació del mòdul es defineix mitjançant l'arxiu XML `menu_views.xml`.

El mòdul crea un menú principal anomenat "**Gestió d'Equips**", sota el qual es disposen diversos submenús organitzats per funcionalitat:

- **Equips**: vinculat al model `gestion.equipo`, permet consultar i gestionar els dispositius disponibles.
- **Préstecs**: vinculat a `gestion.prestamo`, mostra els registres de préstecs actius o passats.
- **Empleats**: relacionat amb `gestion.empleado`, on es gestionen les dades del personal.
- **Cerca amb IA**: accés directe a l'assistent d'intel·ligència artificial (`gestion.búsqueda.ia`) que obre un formulari modal per fer consultes a través de l'API d'OpenAI.

Cada submenú té associada una acció (`ir.actions.act_window`) que especifica el model, el mode de vista (`list, form`) i el comportament de visualització (normal o modal). Aquesta estructura modular permet una navegació clara i jeràrquica des de la interfície d'Odoo, facilitant l'accés ràpid a cada component funcional del mòdul.

### 4.1. Model `gestion.equipo`

El model `gestion.equipo` és el responsable de gestionar els equips informàtics disponibles dins de l'organització per al seu préstec a empleats. Aquest model conté tant les dades tècniques com les dades d'ús dels equips. S'ha creat amb el decorador `models.Model`, indicant com a `_name` el nom tècnic del model i com a `_description` una breu descripció funcional.

#### Camps principals

`name`: Nom de l'equip, obligatori.

`marca i modelo`: Informació del fabricant i model de l'equip.

`numero_serie`: Camp identificador únic que actua com a número de sèrie (requereix que no es repeteixi, per això `unique=True`).



tipo: Tipus d'equip, definit amb un `fields.Selection`. Exemples: portàtil, monitor, teclat...  
estado: Estat actual de l'equip (disponible, en préstec, manteniment, donat de baixa).  
fecha\_alta: Data en què es dona d'alta l'equip (per defecte és la data actual).  
descripcion: Descripció opcional de l'equip.  
imagen: Camp binari per pujar una foto o imatge de l'equip.  
prestamo\_ids: Relació One2many amb els préstecs (`gestion.prestamo`) que ha tingut l'equip.  
display\_name: Camp computat que mostra de forma amigable el nom complet de l'equip

Extres:

Camp computat `display_name`


```
@api.depends('name', 'marca', 'modelo')
def _compute_display_name(self):
    for record in self:
        record.display_name = f"{record.name} ({record.marca or ''} {record.modelo or ''})"
```

Lògica amb `@onchange`

```
@api.onchange('prestamo_ids')
def _onchange_estado(self):
    for record in self:
        if any(p.estado == 'activo' for p in record.prestamo_ids):
            record.estado = 'prestado'
        elif record.estado == 'prestado':
            record.estado = 'disponible'
```

Validacions amb `@api.constrains`

```
@api.constrains('marca', 'modelo', 'numero_serie', 'tipo', 'estado', 'fecha_alta')
def _check_required_fields(self):
    ...
```

 Relació amb préstecs

```
prestamo_ids = fields.One2many('gestion.prestamo', 'equipo_id', string="Historial de Préstecs")
```

equipo.py 1 X

C:\> Program Files > Odoo 18.0.20250127 > server > odoo > addons > gestio\_equipos\_empleats > models > equipo.py > Equipo

```
1 from odoo import models, fields, api, exceptions
2
3 class Equipo(models.Model):
4     _name = 'gestion.equipo'
5     _description = 'Equip informàtic'
6     _rec_name = 'display_name'
7
8     name = fields.Char(string="Nom de l'Equip", required=True)
9     marca = fields.Char(string="Marca", required=True)
10    modelo = fields.Char(string="Model", required=True)
11
12    # ✅ Aquest és el camp que et faltava
13    numero_serie = fields.Char(string="Núm. de Sèrie", required=True, unique=True)
14
15    tipo = fields.Selection([
16        ('portatil', 'Portàtil'),
17        ('monitor', 'Monitor'),
18        ('raton', 'Ratolí'),
19        ('teclado', 'Teclat'),
20        ('auriculars', 'Auriculars'),
21        ('projector', 'Projector'),
22        ('altre', 'Altres')
23    ], string="Tipus d'Equip", default='altre', required=True)
24
25    estado = fields.Selection([
26        ('disponible', 'Disponible'),
27        ('prestado', 'En Prèstec'),
28        ('mantenimiento', 'En Manteniment'),
29        ('baixa', 'Donat de Baixa')
30    ], string="Estat", default='disponible', required=True)
31
32    fecha_alta = fields.Date(string="Data d'Alta", default=fields.Date.today, required=True)
33    descripcion = fields.Text(string="Descripció")
34    imagen = fields.Binary(string="Imatge")
35
36    prestamo_ids = fields.One2many('gestion.prestamo', 'equipo_id', string="Historial de Prèstecs")
```

## 4.2. Model gestion.empleado

El model `gestion.empleado` representa la informació dels empleats que poden rebre equips en préstec. Es tracta d'un model senzill però funcional, amb camps bàsics d'identificació i contacte, així com una relació directa amb els préstecs que ha fet cada empleat.

### Camps principals

name: Nom de l'empleat (obligatori).

cognoms: Cognoms de l'empleat (obligatori).

email: Adreça electrònica de contacte (obligatori).

telefon: Telèfon de contacte (obligatori).

departament: Departament de l'empresa on treballa. És un camp de selecció amb diverses opcions (IT, RRHH, Finances...).

foto: Imatge de l'empleat. S'utilitza per fer més visual la seva fitxa.

prestamo\_ids: Relació One2many amb els préstecs (model gestion.prestamo) associats a aquest empleat.

Aquesta relació permet veure, directament des de la fitxa de l'empleat, quins equips ha tingut en préstec i fer-ne el seguiment històric.

Validació amb `@api.constrains`

```
@api.constrains('name', 'cognoms', 'email', 'telefon', 'departament', 'foto')
def _check_required_fields(self):
    ...
```

Aquesta funció comprova que tots els camps obligatoris estiguin plens. Si no ho estan, llança un error informant quins camps falten. Això assegura que la informació dels empleats sigui coherent i completa, i evita registres buits o mal formats que podrien dificultar la gestió.

### Relació amb préstecs

```
prestamo_ids = fields.One2many('gestion.prestamo', 'empleado_id', string="Préstecs")
```

Aquest camp One2many vincula cada empleat amb tots els préstecs que ha tingut. És molt útil per fer consultes i informes sobre quins equips ha utilitzat cada persona, i forma part de l'arquitectura relacional del sistema.

```

equipo.py 1  empleado.py 1 X
C: > Program Files > Odoo 18.0.20250127 > server > odoo > addons > gestio_equipos_empleats > models > empleado.py > Empleado
1  from odoo import models, fields, api, exceptions
2
3  class Empleado(models.Model):
4      _name = 'gestion.empleado'
5      _description = 'Empleat'
6      _rec_name = 'name'
7
8      name = fields.Char(string="Nom", required=True)
9      cognoms = fields.Char(string="Cognoms", required=True)
10     email = fields.Char(string="Email", required=True)
11     telefon = fields.Char(string="Telèfon", required=True)
12     departament = fields.Selection([
13         ('it', 'Informàtica'),
14         ('rrhh', 'Recursos Humans'),
15         ('finances', 'Finances'),
16         ('logistica', 'Logística'),
17         ('altres', 'Altres')
18     ], string="Departament", default='altres', required=True)
19     foto = fields.Image(string="Foto")
20     prestamo_ids = fields.One2many('gestion.prestamo', 'empleado_id', string="Préstecs")
21
22     @api.constrains('name', 'cognoms', 'email', 'telefon', 'departament', 'foto')
23     def _check_required_fields(self):
24         for record in self:
25             missing = []
26             if not record.name: missing.append("Nom")
27             if not record.cognoms: missing.append("Cognoms")
28             if not record.email: missing.append("Email")
29             if not record.telefon: missing.append("Telèfon")
30             if not record.departament: missing.append("Departament")
31             if not record.foto: missing.append("Foto")
32             if missing:
33                 raise exceptions.ValidationError(f"Falten camps obligatoris: {' '.join(missing)}")
34

```

### 4.3. Model gestion.prestamo

El model `gestion.prestamo` és el cor del mòdul, ja que gestiona els **préstecs d'equips informàtics** als empleats. Aquest model incorpora relacions amb els altres models (`equipo` i `empleado`), així com lògica de negoci automatitzada per facilitar-ne la gestió.

#### Camps i Relacions

`equipo_id`: Relació `Many2one` amb el model `gestion.equipo`, representa l'equip prestat.

`empleado_id`: Relació `Many2one` amb el model `gestion.empleado`, indica quin empleat rep l'equip.

`fecha_inicio`: Data d'inici del préstec (per defecte és avui).

`fecha_fin`: Data en què s'hauria de retornar l'equip.

`estado`: Estat actual del préstec, amb les opcions:

`activo`: Préstec actiu.

`devuelto`: Retornat correctament.

`retrasado`: El termini de devolució ha vençut.

`observacions`: Comentaris addicionals del préstec.

`display_name`: Nom computat que mostra una descripció clara del préstec com per exemple: **"Portàtil Lenovo → Joan García"**.

Validació amb `@api.constrains`

```
@api.depends('equipo_id', 'empleado_id', 'fecha_inicio')
def _compute_display_name(self):
    ...
```

Quan es modifica la `fecha_fin`, s'actualitza automàticament l'`estado`:

```
@api.onchange('fecha_fin')
def check_estado(self):
    ...
```

Botó per Imprimir Informe

El mètode `action_print_report()` permet generar un **informe PDF** del préstec, mitjançant QWeb, que mostra totes les dades del préstec de manera professional.

```
def action_print_report(self):
    return self.env.ref('gestio_equipos_empleats.action_report_prestec').report_action(self)
```

Botó per Enviar per Correu

```
def action_send_report_email(self):  
    ...
```

Aquesta funcionalitat s'ha complementat amb la **configuració SMTP** del servidor de Gmail per fer proves reals d'enviament. explicada posteriorment

## 5. Comprovacions de Dades

Per garantir la **integritat de les dades** i evitar errors habituals en la inserció d'informació, el mòdul implementa **validacions específiques** tant a nivell de camp com a nivell de model.

A continuació es detallen les principals mesures adoptades:

### Validació de camps obligatoris

Tant al model `gestion.empleado` com a `gestion.equipo` s'utilitza el decorador `@api.constrains` per comprovar que l'usuari ha omplert **tots els camps essencials** abans de guardar el registre.

Exemple al model `gestion.empleado`:

```
@api.constrains('name', 'cognoms', 'email', 'telefon', 'departament', 'foto')
def _check_required_fields(self):
    ...
```

Això evita, per exemple, que un usuari pugui crear un empleat sense **nom**, **email** o **telèfon**, o que guardi equips sense la **marca**, el **model** o el **número de sèrie**.

### Validació de valors incoherents

El model `gestion.prestamo` aplica una lògica amb `@api.onchange('fecha_fin')` per determinar automàticament si el préstec és **actiu**, **retornat** o **endarrerit**, en funció de la data actual. Això evita errors humans i manté actualitzat l'estat de forma dinàmica.

### Evitar duplicats: número de sèrie i DNI

Per garantir que cada equip i cada empleat són únics:

- Al model `gestion.equipo`, el camp `numero_serie` està definit com a únic (`unique=True`), de manera que no es poden crear dos equips amb el mateix número.
- Tot i que no s'ha implementat un camp `dni` en el model `gestion.empleado`, es podria afegir amb una restricció similar per garantir unicitat.

Això ajuda a evitar errors de duplicació molt comuns en sistemes de gestió, i assegura que cada registre té una identificació pròpia clara.

## Seguretat extra a nivell d'interfície

Gràcies a les validacions i la configuració de permisos, es **bloqueja la creació o modificació** de registres amb dades incompletes o incorrectes, evitant així que es corrompi la base de dades.

Codi Security.xml

```
<?xml version="1.0" encoding="utf-8"?>
<odoo>
  <data>
    <!-- Grup: Administrador de la Gestió -->
    <record id="group_gestio_equipos_admin" model="res.groups">
      <field name="name">Administrador Gestió</field>
    </record>

    <!-- Grup: Consultor de la Gestió -->
    <record id="group_gestio_equipos_consultor" model="res.groups">
      <field name="name">Consultor Gestió</field>
    </record>
  </data>
</odoo>
```

Codi ir.model.access



























```
> Program Files > Odoo 18.0.20250127 > server > odoo > addons > gestio_equipos_empleats > security > ir.model.access.csv
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 access_equipos_admin,Administrador Equip,model_gestion_equipos,gestio_equipos_empleats.group_gestio_equipos_admin,1,1,1,1
3 access_equipos_consultor,Consultor Equip,model_gestion_equipos,gestio_equipos_empleats.group_gestio_equipos_consultor,1,0,0,0
4 access_prestamo_admin,Administrador Préstec,model_gestion_prestamo,gestio_equipos_empleats.group_gestio_equipos_admin,1,1,1,1
5 access_prestamo_consultor,Consultor Préstec,model_gestion_prestamo,gestio_equipos_empleats.group_gestio_equipos_consultor,1,0,0,0
6 access_empleado_admin,Administrador Empleado,model_gestion_empleado,gestio_equipos_empleats.group_gestio_equipos_admin,1,1,1,1
7 access_empleado_consultor,Consultor Empleado,model_gestion_empleado,gestio_equipos_empleats.group_gestio_equipos_consultor,1,0,0,0
8 access_busqueda_ia_admin,Administrador IA,model_gestion_busqueda_ia,gestio_equipos_empleats.group_gestio_equipos_admin,1,1,1,1
9 access_busqueda_ia_consultor,Consultor IA,model_gestion_busqueda_ia,gestio_equipos_empleats.group_gestio_equipos_consultor,1,1,0,0
10 access_prestamo_mail_wizard_admin,Administrador PrestamoMailWizard,model_gestion_prestamo_mail_wizard,gestio_equipos_empleats.group_gestio_equipos_admin,1,1,1,1
11 access_prestamo_mail_wizard_consultor,Consultor PrestamoMailWizard,model_gestion_prestamo_mail_wizard,gestio_equipos_empleats.group_gestio_equipos_consultor,1,0,0,0
12
13
14
```

Aquesta estructura de permisos permet:

- Que **els administradors** puguin gestionar completament el sistema (alta, modificació i eliminació d'equips, empleats i préstecs).
- Que **els consultors** tinguin un accés limitat i segur, només per consultar dades.



Permisos finals

Funcionalitat	Administrador Gestió 	Consultor Gestió 
Veure equips	 Sí	 Sí
Crear un equip nou	 Sí	 No
Editar informació d'un equip	 Sí	 No
Eliminar un equip	 Sí	 No
Veure préstecs	 Sí	 Sí
Crear un préstec nou	 Sí	 No
Modificar l'estat d'un préstec	 Sí	 No
Esborrar un préstec	 Sí	 No
Veure empleats	 Sí	 Sí
Afegir o editar un empleat	 Sí	 No
Eliminar empleats	 Sí	 No
Fer cerques amb IA (Cerca amb IA)	 Sí	 Sí (si té permisos d'accés al model)

## 6. Impressió d'Informes PDF

S'ha desenvolupat una funcionalitat completa per **generar informes en format PDF** mitjançant el motor de plantilles **QWeb** d'Odoo, associada al model `gestion.prestamo`. Aquesta funcionalitat permet obtenir un **informe visual i professional** per a cada préstec realitzat.

### Estructura tècnica implementada

#### Fitxer XML de la plantilla QWeb (`prestamo_report_template`):

Utilitza el `t-call="web.basic_layout"` per aprofitar la maquetació base d'Odoo. S'utilitza un `t-foreach` per recorre cada registre del model `gestion.prestamo` i generar la seva informació.

### Contingut i estil de l'informe

L'informe conté:

**Logotip personalitzat del mòdul:** S'ha afegit la imatge de `/static/description/icon.png` a la part superior de l'informe.

**Títol centrat:** "Informe de Préstec" amb estil net i destacat.

**Taula de dades** amb estil modern, mostrant:

- Equip prestat (`equipo_id`)
- Empleat (`empleado_id`)
- Data d'inici i de retorn
- Estat del préstec (actiu, retornat o endarrerit)
- Observacions (amb valor per defecte '—' si està buit)

**Data de generació automàtica** (`create_date`) formatada correctament.

**Camp per signatura** de l'empleat, simulant un document físic

## Declaració de l'acció d'informe

```
<record id="action_report_prestec" model="ir.actions.report">
  <field name="name">Informe de Préstec</field>
  <field name="model">gestion.prestamo</field>
  <field name="report_type">qweb-pdf</field>
  <field name="report_name">gestio_equipos_empleats.prestamo_report_template</field>
  <field name="report_file">gestio_equipos_empleats.prestamo_report_template</field>
  <field name="print_report_name">'Prestec_' + object.display_name</field>
</record>
```

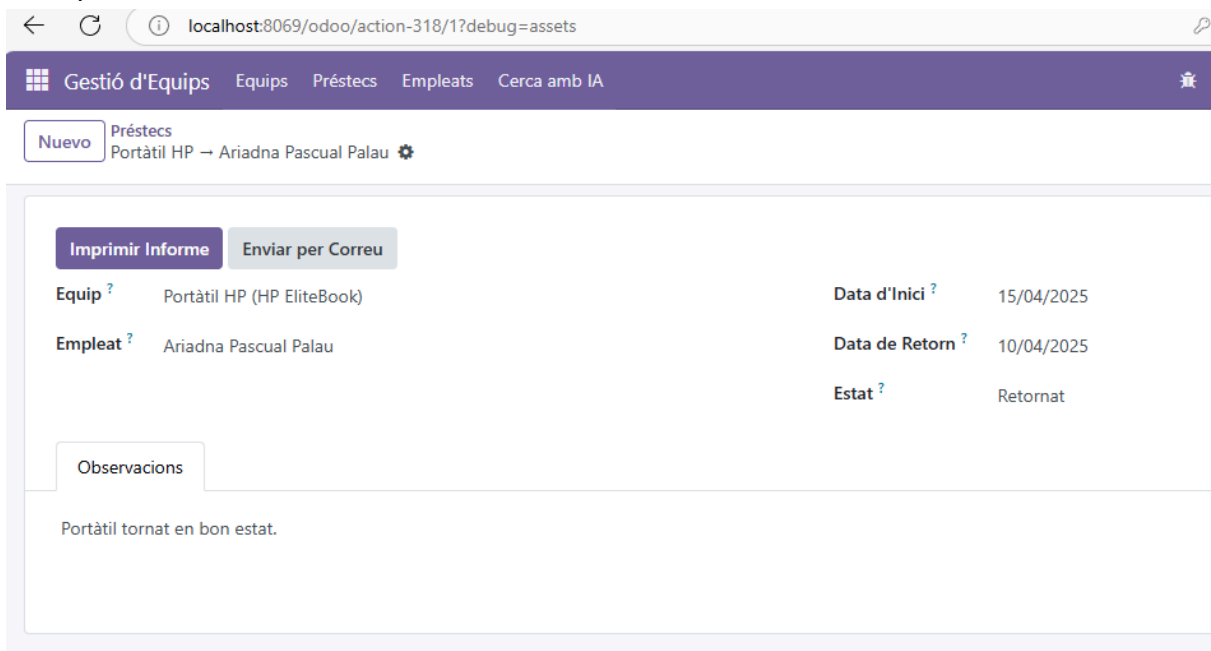
Crida des del botó "Imprimir Informe"

Al model Python (`gestion.prestamo`) s'ha afegit una funció `action_print_report` que crida aquesta acció declarada:

```
def action_print_report(self):
    return self.env.ref('gestio_equipos_empleats.action_report_prestec').report_action(self)
```

Aquest mètode permet que **en fer clic al botó "Imprimir Informe"**, es generi el PDF automàticament amb les dades del registre actiu.

## Exemple



The screenshot shows the Odoo web interface for a laptop record. The breadcrumb trail is: **Gestió d'Equips** > **Equips** > **Préstecs** > **Empleats** > **Cerca amb IA**. The record is titled "Nuevo Préstecs" and shows "Portàtil HP → Ariadna Pascual Palau".

At the top of the record form, there are two buttons: **Imprimir Informe** (highlighted in purple) and **Enviar per Correu** (grey).

The record details are as follows:

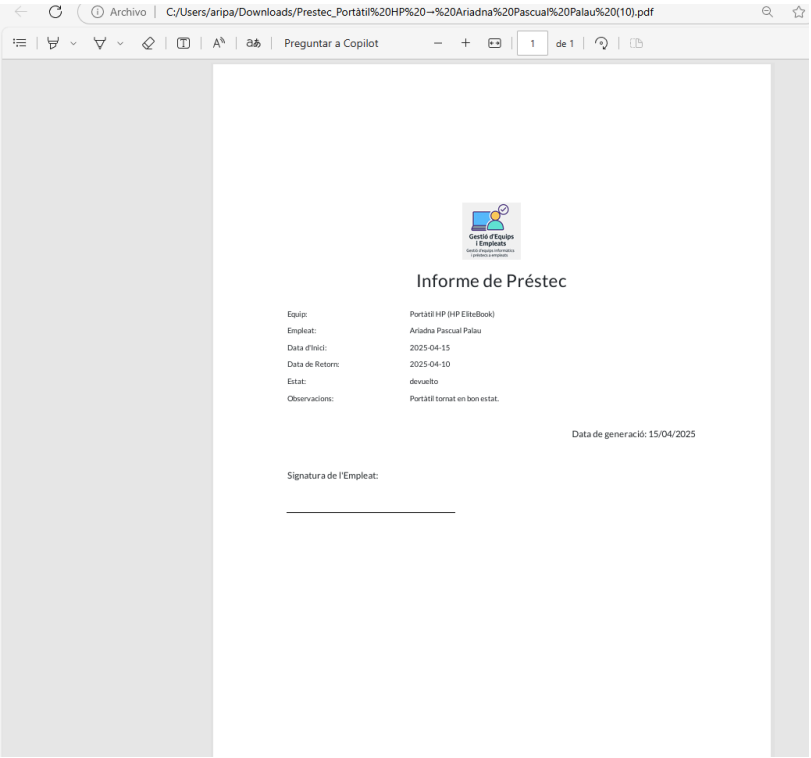
Equip ?	Data d'Inici ?
Portàtil HP (HP EliteBook)	15/04/2025

Empleat ?	Data de Retorn ?
Ariadna Pascual Palau	10/04/2025

Below these, there is a field for **Estat ?** with the value "Retornat".

At the bottom, there is a tab labeled **Observacions**. The text inside the tab reads: "Portàtil tornat en bon estat."

Resultat del botó d'imprimir:



Descargar i imprimir 😊

## 7. Enviament per Correu

S'ha implementat un sistema complet per **enviar l'informe PDF dels préstecs per correu electrònic** a través d'un *wizard* basat en `TransientModel`.

Mòdul Transient: `gestion.prestamo.mail.wizard`

```
class PrestamoMailWizard(models.TransientModel):
    _name = 'gestion.prestamo.mail.wizard'
    _description = 'Assistent per Enviar Informe de Préstec per Correu'
```

Aquest model conté els camps:

`email_to`: Adreça electrònica del destinatari.

`message_body`: Cos del missatge (editable).

`prestamo_id`: Enllaç amb el préstec associat.

📧 Funció de lògica `send_mail_with_report()`

```
def send_mail_with_report(self):
    if not self.prestamo_id:
        raise UserError("No s'ha trobat cap préstec associat.")

    report = self.env.ref('gestio_equipos_empleats.action_report_prestec')
    pdf_content, _ = report.render_qweb_pdf(report.id, res_ids=[self.prestamo_id.id])

    attachment = self.env['ir.attachment'].create({
        'name': f"Informe_Prestec_{self.prestamo_id.display_name}.pdf",
        'type': 'binary',
        'datas': base64.b64encode(pdf_content),
        'res_model': 'gestion.prestamo',
        'res_id': self.prestamo_id.id,
        'mimetype': 'application/pdf',
    })

    mail_values = {
        'subject': f"Informe de Préstec - {self.prestamo_id.display_name}",
        'body_html': f"<p>{self.message_body}</p>",
        'email_to': self.email_to,
        'email_from': self.env.user.email or 'no-reply@example.com',
        'attachment_ids': [(6, 0, [attachment.id])],
    }

    mail = self.env['mail.mail'].create(mail_values)
    mail.send()

    # ✨ Mostrar notificació de èxit
    return {
        'type': 'ir.actions.client',
        'tag': 'display_notification',
        'params': {
            'title': 'Èxit',
            'message': 'El correu s\'ha enviat correctament!',
            'type': 'success',
            'sticky': False,
        }
    }
```

Aquesta funció:

**Comprova que hi ha un préstec associat.**

**Renderitza el PDF** amb `_render_qweb_pdf()` usant l'informe personalitzat de `gestion.prestamo`.

**Crea un fitxer adjunt** (`ir.attachment`) amb el PDF generat.

**Crea i envia el correu** amb `mail.mail.create()` i el mètode `.send()`.

Mostra una **notificació d'èxit** a l'usuari.

## SERVIDOR CONFIGURAT

Per permetre l'enviament de correus, s'ha configurat correctament un servidor de correu sortint SMTP. Concretament:

Proveïdor: Gmail

Tipus de seguretat: TLS

Port: 587

Autenticació: Correu de Gmail amb contrasenya d'aplicació generada (no la contrasenya habitual).

Aquesta configuració es realitza des del panell de Configuració > General > Servidors de Correu de Sortida.

The screenshot shows the 'Ajustes / Servidores de correo de salida' (Settings / Outgoing Email Servers) page in Odoo. The page is titled 'Probar conexión' (Test connection). It contains a form for configuring an outgoing email server. The form is divided into two main sections: 'Nombre' (Name) and 'Conexión' (Connection). The 'Nombre' section includes fields for 'Nombre' (set to 'EnviarInforme'), 'Prioridad' (set to 10), 'Filtro DESDE' (set to 'p. ej. email@dominio.com, dominio.com'), and 'Autenticar con' (set to 'Nombre de usuario'). The 'Conexión' section includes fields for 'Encriptación de la conexión' (set to 'TLS (STARTTLS)'), 'Nombre de usuario' (set to 'ari.palau20@gmail.com'), 'Contraseña' (masked with dots), 'Depurando' (unchecked), 'Servidor SMTP' (set to 'smtp.gmail.com'), and 'Puerto SMTP' (set to 587). There is a 'Detectar límite máximo' button and a note about converting attachments to links for emails.

Nombre		Prioridad
Nombre	EnviarInforme	10
Filtro DESDE	p. ej. email@dominio.com, dominio.com	
Autenticar con	<input checked="" type="radio"/> Nombre de usuario <input type="radio"/> Certificado SSL <input type="radio"/> Interfaz de línea de comandos <input type="radio"/> Autenticación OAuth de Gmail	

Conexión	
Encriptación de la conexión	<input type="radio"/> Ninguno <input checked="" type="radio"/> TLS (STARTTLS) <input type="radio"/> SSL/TLS
Nombre de usuario	ari.palau20@gmail.com
Contraseña	.....
Depurando	<input type="checkbox"/>
Servidor SMTP	smtp.gmail.com
Puerto SMTP	587

Vista Assistent

```

<form string="Enviar Informe per Correu">
  <group>
    <field name="email_to"/>
    <field name="message_body" widget="text"/>
    <field name="prestamo_id" invisible="1"/>
  </group>
  <footer>
    <button string="Enviar" type="object" name="send_mail_with_report" class="btn-primary"/>
    <button string="Cancel·lar" special="cancel" class="btn-secondary"/>
  </footer>
</form>

```

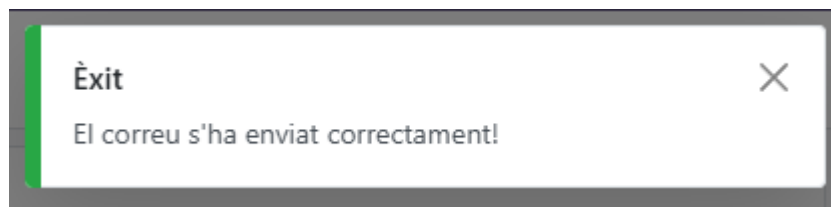
## RESULTAT AMB NOTIFICACIÓ DE ÈXIT

Enviar Informe per Correu

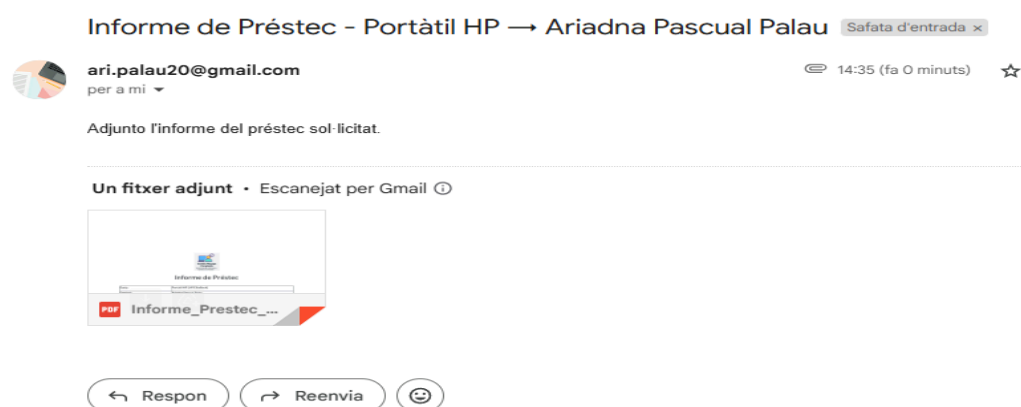
Correu Destinatari ? ari.palau20@gmail.com

Missatge ? Adjunto l'informe del préstec sol·licitat.

Enviar Cancel·lar



## CORREU ENVIAT:



## 8. Assistència IA (OpenAI)

S'ha integrat en el mòdul una funcionalitat innovadora que permet fer **consultes directament a una Intel·ligència Artificial (IA)** basada en l'API d'OpenAI.

Aquesta funcionalitat ofereix una **ajuda contextual dins de l'entorn d'Odoo**, ideal per a donar suport als usuaris en tasques relacionades amb equips, tecnologia, preguntes generals, etc.

### Transient Model: `gestion.busqueda.ia`

Per gestionar les consultes, s'ha creat un model temporal (`TransientModel`) anomenat `gestion.busqueda.ia`, que conté els següents camps:

- **pregunta**: camp de text on l'usuari escriu la seva consulta.
- **respuesta**: resposta retornada per la IA, mostrada de forma **només lectura**.

```
import requests
from odoo import models, fields

class BusquedaIA(models.TransientModel):
    _name = 'gestion.busqueda.ia'
    _description = 'Buscar informació amb IA'

    pregunta = fields.Text(string="Pregunta")
    respuesta = fields.Text(string="Resposta", readonly=True)

    def buscar_con_ia(self):
        try:
            api_key = "sk-proj-8UUSlWdcIcT3BWIOZopNRxQWqF8KtyMoA1sZaNQ2XonFARN7-wle5d9drz5SBdDT4-ZnxytPT3BlbkfJMDp-9juJZSa7ewwIk6LfgLWMnGxcLwxyexSGexsnBN9KG9JItlyfem5er"
            url = "https://api.openai.com/v1/chat/completions"

            headers = {
                "Authorization": f"Bearer {api_key}",
                "Content-Type": "application/json",
            }

            data = {
                "model": "gpt-3.5-turbo",
                "messages": [
                    {"role": "system", "content": "Ets un assistent útil."},
                    {"role": "user", "content": self.pregunta},
                ]
            }

            response = requests.post(url, headers=headers, json=data, timeout=20)

            if response.status_code == 200:
                result = response.json()
                self.respuesta = result['choices'][0]['message']['content']
            else:
                self.respuesta = f"Error {response.status_code}: {response.text}"

        except requests.exceptions.RequestException as e:
```

### Vista XML del Formulari

S'ha creat una vista de formulari molt senzilla però funcional, amb els següents elements:

- Un **camp per escriure la pregunta**.
- Un **camp en readonly** per mostrar la resposta de la IA.
- Botons de control:
  - “Buscar” per enviar la petició a OpenAI.



- “Cancel·lar” per tancar el modal.

Aquesta vista s'executa com una finestra modal (`target="new"`) per facilitar una experiència fluida i no intrusiva.

### Com fem la crida?

La consulta a la IA es realitza amb una crida POST a l'endpoint oficial de l'API d'OpenAI:

`https://api.openai.com/v1/chat/completions`

El model utilitzat és `gpt-3.5-turbo`.

Es defineix un **context de sistema** ("Ets un assistent útil.") i es passa la pregunta de l'usuari.

El resultat es desa automàticament en el camp `respuesta`.

La connexió inclou tractament d'errors per:

- Errors de xarxa (timeout, connexions fallides).
- Errors de resposta del servidor.
- Errors imprevistos (excepcions generals).

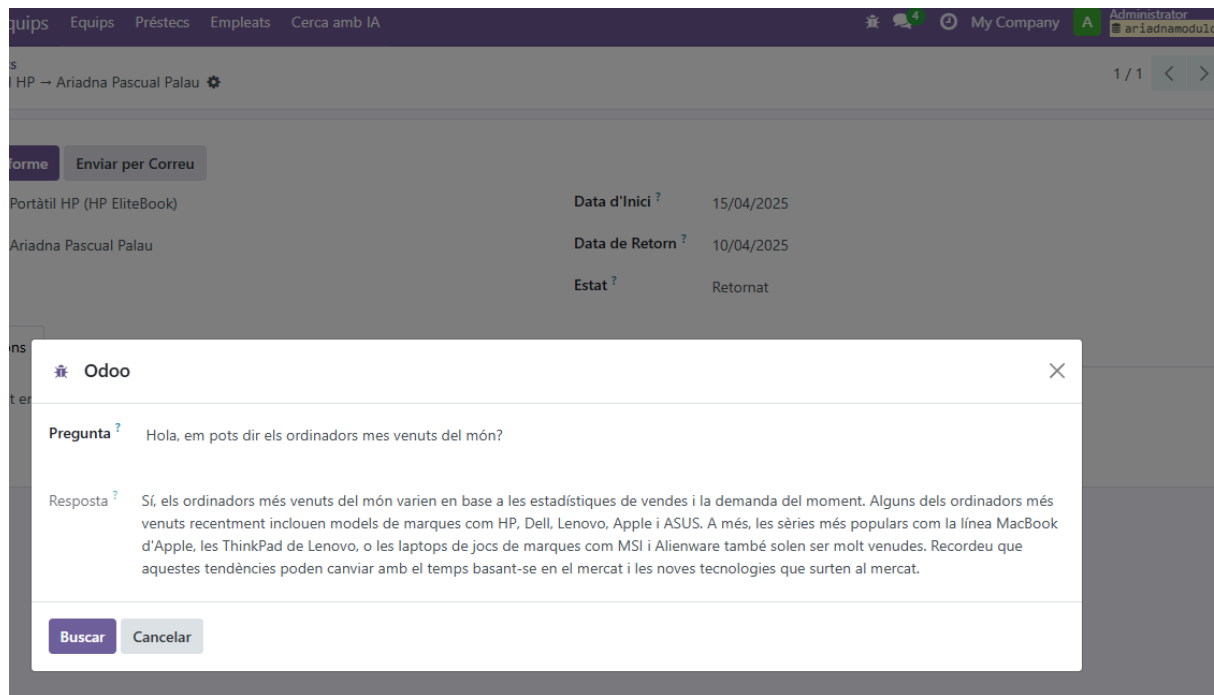
### Autenticació i Clau API

L'autenticació es fa mitjançant una **clau d'API personal d'OpenAI**, especificada dins el codi Python. Aquesta clau pot ser:

- **Definida directament** dins del codi (no recomanat en producció).
- O gestionada a través de paràmetres de sistema o fitxers `.env` per més seguretat.

```
headers = {  
    "Authorization": f"Bearer {api_key}",  
    "Content-Type": "application/json",  
}
```

Exemple:



Aquesta funcionalitat ofereix un **valor afegit molt potent** per al mòdul, ja que permet integrar AI per ajudar a:

- Donar suport tècnic.
- Resoldre dubtes d'ús.
- Proporcionar respostes contextuais directament dins Odoo.

## 9. Dades de Demostració

Per tal de provar i mostrar la funcionalitat del mòdul de forma immediata després d'instal·lar-lo, s'ha creat un fitxer `demo.xml` dins la carpeta `data/`. Aquest fitxer inclou dades de mostra per a les entitats principals del sistema.

El fitxer `demo.xml` conté:

- Un equip exemple amb sèrie, descripció i imatge.
- Un empleat amb totes les dades completes.
- Un préstec de demostració relacionat.

```
<odoo>
<!-- Demo Equip -->
<record id="equipo_portatil_1" model="gestion.equipo">
  <field name="name">Portàtil HP</field>
  <field name="marca">HP</field>
  <field name="modelo">EliteBook</field>
  <field name="tipo">portatil</field>
  <field name="estado">disponible</field>
  <field name="fecha_alta">2024-01-15</field>
</record>

<record id="equipo_monitor_1" model="gestion.equipo">
  <field name="name">Monitor LG</field>
  <field name="marca">LG</field>
  <field name="modelo">Ultrawide</field>
  <field name="tipo">monitor</field>
  <field name="estado">disponible</field>
  <field name="fecha_alta">2024-02-10</field>
</record>

<!-- Demo Préstec -->
<record id="prestamo_demo_1" model="gestion.prestamo">
  <field name="equipo_id" ref="equipo_portatil_1"/>
  <field name="empleado">Ariadna Pascual</field>
  <field name="fecha_inicio">2024-04-01</field>
  <field name="fecha_fin">2024-04-10</field>
  <field name="estado">devuelto</field>
  <field name="observacions">Portàtil tornat en bon estat.</field>
</record>
</odoo>
```

### Ús de les Dades Demo

Aquestes dades s'inclouen en el `__manifest__.py` del mòdul mitjançant:





```
'demo': ['data/demo.xml'],
```

## 10. Perfils d'Usuari i Permisos

Per garantir la seguretat i el control d'accés dins l'aplicació, s'han creat **dos grups d'usuari diferenciats**:





### 1. Administrador (**group\_gestio\_equips\_admin**)

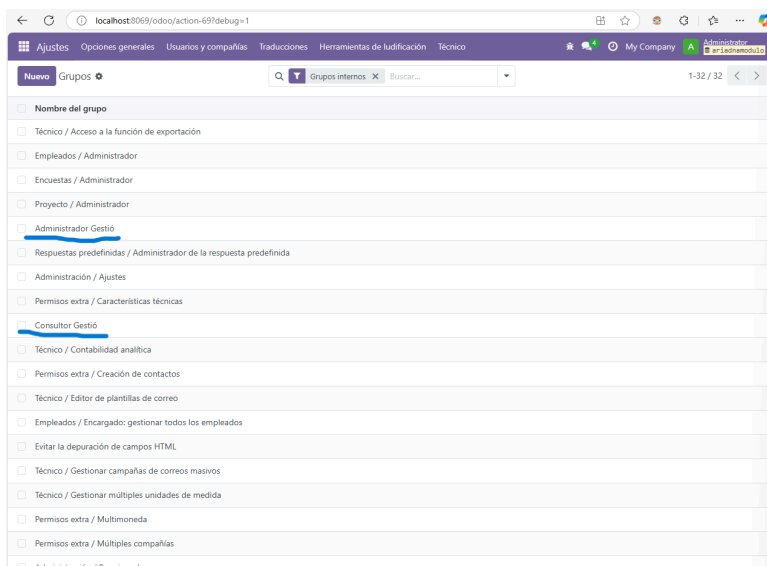
Aquest grup està pensat per a l'usuari responsable de la gestió completa del sistema. Té **permís total** sobre tots els models:

- Lectura 
- Escriptura 
- Creació 
- Eliminació 

### 2. Consultor (**group\_gestio\_equips\_consultor**)

Aquest rol està dissenyat per a usuaris amb accés **només consultiu**. Pot veure dades, però **no pot modificar-les**:

- Lectura 
- Escriptura 
- Creació 
- Eliminació 



♦ ir.model.access.csv

```
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_equipo_admin,Administrador
Equip,model_gestion_equip,gestio_equipos_empleats.group_gestio_equipos_admin,1,1,1,1
access_equipo_consultor,Consultor
Equip,model_gestion_equip,gestio_equipos_empleats.group_gestio_equipos_consultor,1,0,0,0
```

Aquesta estructura es repeteix per als següents models:

gestion.equipo

gestion.empleado

gestion.prestamo

gestion.busqueda.ia

gestion.prestamo.mail.wizard

## 11. Assignació de Grups a Usuaris

Un cop creats els grups i els permisos d'accés, es poden assignar fàcilment des de la interfície d'Odoo:

1. Ves a **Ajustes > Usuarios**.
2. Selecciona el teu usuari (ex: *Ariadna Pascual*).
3. A la pestanya **Altres permisos**, marca un dels grups següents:
  - ♦ *Gestor de Préstecs* (administrador complet)
  - ♦ *Consultor de Préstecs* (només lectura)

Aquesta assignació permet personalitzar els accessos per usuari i **garantir que només puguin veure o editar segons el seu rol**.

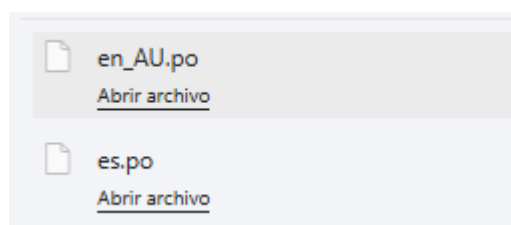
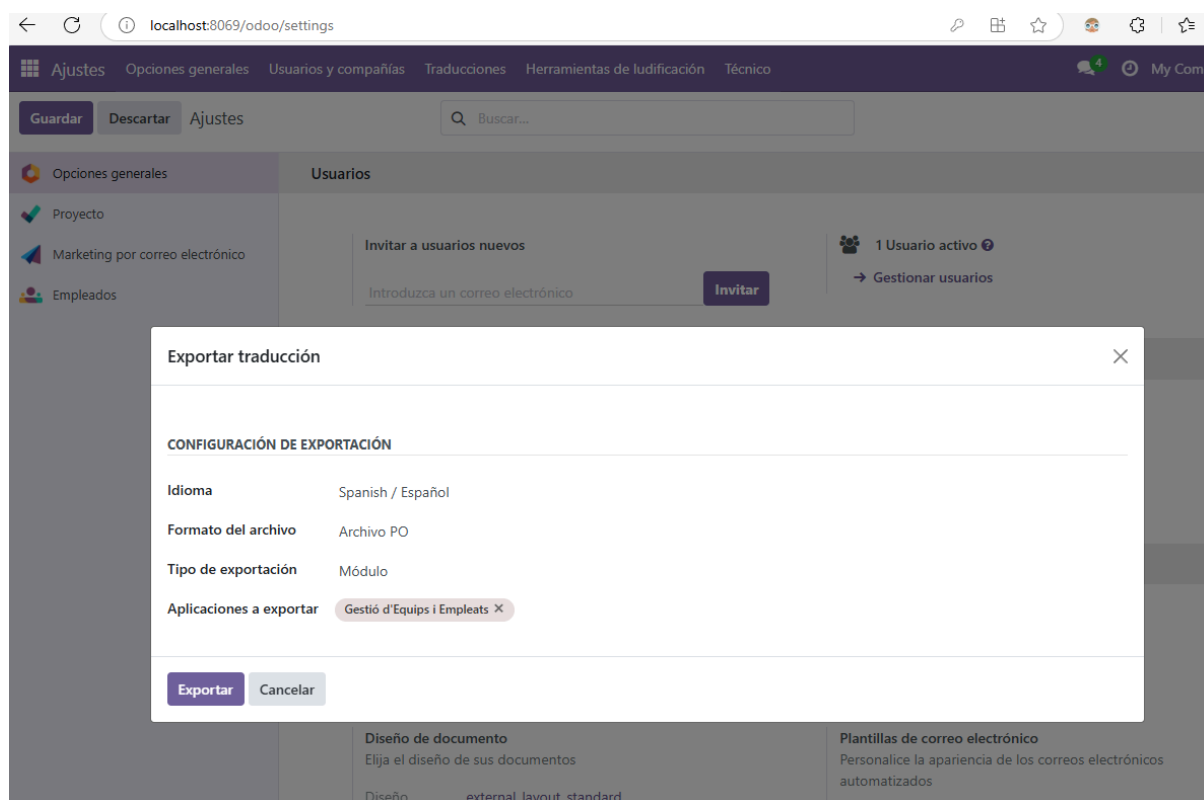
## 12. Traducció amb Poedit

Amb l'objectiu de fer el mòdul accessible per a usuaris de diferents idiomes, s'ha iniciat la traducció del mòdul "**Gestió d'Equips i Empleats**" utilitzant l'eina **Poedit**, basada en fitxers **.po**, el format estàndard de traducció de cadenes a Odoo.

### Estructura de Traducció

Per defecte, Odoo carrega les traduccions des de la carpeta:

`/gestio_equipos_empleados/i18n/`



```
Archivo  Editor  Ver
"Content-Transfer-Encoding: \n"
"Plural-Forms: \n"

#. module: gestio_equipos_empleats
#: model:ir.actions.report,print_report_name:gestio_equipos_empleats.action_report_prestec
msgid "'Prestec_' + object.display_name"
msgstr ""

#. module: gestio_equipos_empleats
#: model_terms:ir.ui.view,arch_db:gestio_equipos_empleats.prestamo_report_template
msgid "<strong>Data de generació:</strong>"
msgstr ""

#. module: gestio_equipos_empleats
#: model_terms:ir.ui.view,arch_db:gestio_equipos_empleats.prestamo_report_template
msgid "<strong>Signatura de l'Empleat:</strong>"
msgstr ""

#. module: gestio_equipos_empleats
#:
model:ir.model.fields.selection,name:gestio_equipos_empleats.selection__gestion_prestamo_estado
__activo
msgid "Actiu"
msgstr ""

#. module: gestio_equipos_empleats
#: model_terms:ir.ui.view,arch_db:gestio_equipos_empleats.prestamo_report_template
msgid "Data de generació: "
msgstr ""

Ln 13, Col 22 | 17.882 caracteres | 100% | Unix (LF) | UTF-8
```



# FUNCIONALITATS EXTRES

## 13. Firma digital del Treballador

Amb l'objectiu de fer els informes més oficials i amb validesa documental, s'ha afegit la funcionalitat de **signatura digital** dins del model gestion.prestamo. Aquesta millora permet als empleats **signar els informes dels préstecs** directament des de la interfície d'Odoo, mitjançant un widget gràfic.

Es pot escriure directament al camp o pujar una foto d'una firma com el cas següent.

### Implementació tècnica

#### Nou camp al model:

Al model gestion.prestamo s'ha afegit el camp següent

```
signature = fields.Binary(string="Signatura")
```

#### Widget de signatura al formulari:

A la vista XML del formulari de préstecs (view\_prestamo\_form), s'ha afegit el widget signature que permet signar utilitzant el ratolí o el dit (si és pantalla tàctil):

```
<field name="signature" widget="signature"/>
```

#### Integració a l'informe QWeb:

A l'informe prestamo\_report\_template, s'ha modificat el bloc de signatura perquè, si hi ha signatura guardada, es mostri dins el PDF:

```
<t t-if="prestec.signature">
    
</t>
<t t-else="">
    <div style="border-bottom: 1px solid #000; height: 40px; width: 300px;"/>
</t>
```

Resultat:

Gestió d'EquipsEquipsPréstecsEmpleatsCerca amb IA

Nuevo

Préstecs

Portàtil HP → Ariadna Pascual Palau

Imprimir Informe

Enviar per Correu


Equip

Portàtil HP (HP EliteBook)

Empleat

Ariadna Pascual Palau

Signatura



Observacions

Portàtil tornat en bon estat.

Data d'Inici

15/04/2025

Data de Retorn

10/04/2025

Estat

Retornat



# Informe de Préstec

Equip:

Portàtil HP (HP EliteBook)

Empleat:

Ariadna Pascual Palau

Data d'Inici:

2025-04-15

Data de Retorn:

2025-04-10

Estat:

devuelto

Observacions:

Portàtil tornat en bon estat.

Data de generació: 15/04/2025

Signatura de l'Empleat:



## 14. Problemes i Millores

### Problemes Trobats i Solucionats

Durant el desenvolupament del mòdul *Gestió d'Equips i Empleats*, s'han identificat i resolt diversos problemes tècnics que van aparèixer en les diferents fases:

#### Error en imprimir l'informe PDF

En intentar generar l'informe amb `env.ref('gestio_equipos_empleats.action_report_prestec')`, Odoo retornava un error `External ID not found`. Després de revisar el codi, es va detectar que l'ID del report estava mal referenciat o bé el fitxer XML del report no estava carregat correctament.



**Solució:** S'assegura que l'ID declarat en el `<record id="...">` del report coincideixi exactament amb el que es fa servir a `env.ref(...)`, i que el fitxer estigui inclòs al manifest `__manifest__.py`.

#### Error a la referència del report

En una versió prèvia, el nom de la plantilla de report havia canviat (`prestamo_report_template` vs `prestec_report_template`) i no s'havia actualitzat la referència a tot arreu, causant errors d'execució en generar l'informe.

**Solució:** Es van revisar i unificar totes les referències perquè utilitzessin el mateix nom d'identificador (`gestio_equipos_empleats.prestamo_report_template`).

#### Encoding de caràcters especials (emojis)

Inicialment es van utilitzar emojis com , , etc. dins del QWeb. Tot i que es visualitzaven bé en HTML, el generador de PDF (`wkhtmltopdf`) **no era compatible amb tots els emojis**, generant errors o sortides buides.

**Solució:** Es van substituir per text pla o caràcters compatibles per garantir la compatibilitat amb el renderitzat PDF.


#### Menús que no es mostraven

En crear els menús personalitzats (`menu_views.xml`), aquests no apareixien al frontend. El problema es devia a la **manca de permisos d'accés** o al fet que el navegador no havia carregat bé els canvis.

**Solució:** S'assignaren permisos als grups corresponents i es va actualitzar la pàgina amb F5 després de cada modificació.

## Possibles millores del mòdul

Encara que el mòdul funciona correctament, hi ha funcionalitats addicionals que es podrien implementar per millorar l'experiència:

- **Arxiu d'equips donats de baixa:** Actualment no hi ha un filtre específic o vista separada per consultar equips en estat "baixa". Es podria afegir una acció intel·ligent o filtre.
- **Signatura digital:** A l'informe de préstec, hi ha un espai per a la signatura. Es podria implementar un widget per capturar la signatura directament des del frontend (amb canvas).
-  **IA contextual:** Actualment, l'assistent IA respon preguntes generals. Es podria millorar per fer-lo més contextual, per exemple, suggerint equipaments segons el departament.

## 15. Conclusions

Aquest projecte ha permès desenvolupar i desplegar un mòdul completament funcional dins l'entorn Odoo, assolint tots els requisits tant tècnics com funcionals establerts. S'han dissenyat i implementat models ben estructurats, amb relacions adequades entre entitats com equips, empleats i préstecs, assegurant la integritat de les dades i la seva gestió eficaç.

A nivell d'interfície, s'ha proporcionat una experiència d'usuari clara i intuïtiva, amb menús personalitzats, vistes adaptades i accés estructurat segons rols d'usuari (administrador i consultor), garantint així la seguretat i el control de les accions disponibles per cada perfil.

S'han integrat funcionalitats avançades com:

- **Impressió d'informes PDF** amb QWeb, amb disseny visual personalitzat.
- **Enviament d'informes per correu electrònic**, configurant correctament un servidor SMTP (Gmail).
- **Funcionalitat d'assistència intel·ligent** via OpenAI, permetent consultes automatitzades des del propi mòdul.
- **Traducció del mòdul** mitjançant fitxers `.po`, millorant la internacionalització i preparant el projecte per a entorns multilingües.

A més, s'han introduït dades de demostració per validar el funcionament global i s'han aplicat validacions i restriccions per assegurar la qualitat de la informació introduïda.

Aquest projecte no només ha servit per consolidar coneixements sobre la plataforma Odoo, sinó també per aplicar bones pràctiques de desenvolupament modular, escalable i mantenible. Ha suposat una immersió completa tant al backend com a la personalització del frontend, i ha obert la porta a futures ampliacions com la sincronització amb altres sistemes, noves funcionalitats o l'adopció de noves tecnologies.

**En definitiva, s'ha creat un mòdul sòlid, professional i totalment operatiu, preparat per ser utilitzat i evolucionat en entorns reals.**



GRÀCIES

