

# **Práctica 3 - Primeros módulos Odoo**

Sistemas de Gestión Empresarial

Cristian Fernández

15 de noviembre de 2025

# Índice

1. Introducción	2
2. Implantación de módulo básico <i>hola mundo</i>	3
3. Implantación de primer módulo <i>lista de tareas</i>	4
4. Modificación del primer módulo <i>lista de tareas</i>	5
5. Historias de usuario	6
6. Herramientas de desarrollo	7
7. Lenguajes de programación	8
8. Interfaz de la aplicación	9

# 1. Introducción

El objetivo de esta práctica consiste en implantar y configurar los módulos de Odoo indicados en esta misma documentación, demostrando su funcionamiento mediante capturas de pantalla. Además, se realizarán las modificaciones sobre el modelo y la vista del módulo *Lista de tareas* con las mejoras que se consideren oportunas, documentando todo el proceso con capturas.

Se reutilizará el repositorio creado en la práctica anterior, con objeto de tener todo el trabajo bien organizado y disponible para una posible visualización en cualquier momento.

La finalidad de la práctica en detalle es la siguiente:

- **Implantación de Módulo básico.**

- Desarrollar e instalar el módulo *hola mundo* propuesto en el temario.

- **Implantación de Primer módulo.**

- Desarrollar e instalar el módulo *lista de tareas* propuesto en el temario.

- **Modificación de Primer módulo.**

- Modificar el módulo *lista de tareas* con las mejoras que se consideren oportunas y documentar dichas mejoras.

Cada punto tendrá una explicación breve del proceso de creación y se demostrará su funcionamiento mediante capturas. Si fuese necesario también se comentarán los problemas encontrados y las soluciones empleadas.

## 2. Implantación de módulo básico *hola mundo*

Odoo permite la creación de módulos personalizados al gusto del usuario en los que puede extender funcionalidades según se requiera. Para demostrarlo crearemos un módulo de ejemplo básico muy sencillo cuya única finalidad será aparecer listado en la lista de módulos de nuestro contenedor Odoo. El propósito es que el sistema detecte y muestre correctamente el módulo, lo cual nos permitirá avanzar al siguiente punto.

Asentada la premisa comenzaremos la tarea creando una carpeta dentro de nuestro proyecto llamada *Ejemplo01-HolaMundo*, en el interior del directorio en el que hayamos configurado el alojamiento de nuestros módulos personalizados (en mi caso la carpeta *addons*).

La carpeta recién creada contendrá los siguientes ficheros:

- **Fichero *\_\_init\_\_.py*:** Este fichero estará vacío.
- **Fichero *\_\_manifest\_\_.py*:** Este fichero contendrá una línea de código mostrada en próximas capturas.

### 3. Implantación de primer módulo *lista de tareas*

Odoo permite la creación de módulos personalizados al gusto del usuario en los que puede extender funcionalidades según se requiera.

**RNF-1:** El tiempo de respuesta al pulsar el botón Iniciar no debe superar los 2 segundos.

**RNF-2:** La interfaz gráfica de usuario ha de ser fácilmente entendible.

**RNF-3:** Cada acción realizada por los hilos debe ser lo suficientemente descriptiva para el usuario.

**RNF-4:** La interfaz debe avisar de forma clara que la ejecución de la aplicación ha terminado.

## 4. Modificación del primer módulo *lista de tareas*

A continuación se muestran los casos de uso mediante la siguiente tabla:

Caso de uso	Descripción
Consultar leyenda colores	El usuario consulta el funcionamiento de los hilos de la aplicación.
Iniciar aplicación	El usuario puede iniciar la aplicación pulsando un botón.

## 5. Historias de usuario

A continuación se muestran las historias de usuario mediante la siguiente tabla:

ID	Como...	Quiero...	Para...
HU-01	Usuario	Pulsar un botón	Iniciar la aplicación
HU-02	Usuario	Entender el comportamiento de los hilos	Deducir cómo funciona la concurrencia

## 6. Herramientas de desarrollo

**IntelliJ Community Edition** es un IDE gratuito y de código abierto, ideal para el desarrollo en lenguajes JVM y Android. Sus características principales incluyen un potente editor de código con autocompletado inteligente y refactorización, soporte para control de versiones, un depurador integrado, y herramientas de pruebas unitarias. Está disponible para todas las plataformas.

Características principales:

- **Editor de código potente:** Ofrece resaltado de sintaxis, análisis en tiempo real, sugerencias de código, e inspecciones y correcciones rápidas.
- **Soporte para múltiples lenguajes:** Es compatible con Java, Kotlin, Groovy, Scala, y otros lenguajes JVM.
- **Integración con control de versiones:** Permite la integración con sistemas como Git, SVN, y Mercurial.

## 7. Lenguajes de programación

**JavaFX** se caracteriza por su capacidad de crear aplicaciones enriquecidas visualmente, integrando multimedia (audio, video) y gráficos vectoriales. Permite desarrollar aplicaciones multiplataforma (escritorio, web, móvil, smart TV). Otras características clave son la integración fluida con el ecosistema Java y herramientas como FXML y Scene Builder para un desarrollo más eficiente y colaborativo.

Características principales:

- **Interfaces gráficas de usuario (GUI) enriquecidas:** Permite crear interfaces modernas y atractivas con animaciones, gráficos vectoriales, y elementos personalizables.
- **Integración multimedia:** Facilita la inclusión de contenido de audio, video y otros medios dentro de las aplicaciones.
- **Desarrollo declarativo (FXML):** Utiliza FXML, un lenguaje basado en XML, para describir la interfaz de usuario, lo que facilita el trabajo colaborativo entre diseñadores y desarrolladores.

## **8. Interfaz de la aplicación**

A continuación unas capturas de las diferentes vistas de la interfaz.