|  |  |  |
| --- | --- | --- |
| sistemas de gestión | | TAREA 1 UD2 |
|  | | |
|  | | |
|  |  | |
| alumno cesur 25/26  Alejandro Muñoz de la Sierra | PROFESOR  Óscar González Núñez | |

introduccion

En este caso práctico nos pusimos manos a la obra para implementar y desplegar Odoo Community en una pyme del sector logístico. La idea no era simplemente instalar un software, sino conectar de verdad los departamentos de Ventas, Inventario y Clientes para que todo el negocio funcione como un reloj, gestionado desde un único lugar seguro y sin dolores de cabeza.

Para lograrlo, montamos el servidor sobre Windows 11 y definimos quién hace qué, creando roles específicos para ventas, almacén y contabilidad. Luego vino la prueba de fuego: simular el día a día real. Pasamos por todo el ciclo, desde que entra un pedido de venta hasta que se valida la entrega y se actualiza el stock automáticamente, asegurándonos de que cada paso fluyera sin interrupciones.

Claro que en el camino nos encontramos con esas típicas piedras técnicas. Tuvimos que pelear un poco con dependencias externas como Wkhtmltopdf o la librería phonenumbers, y también nos dimos cuenta de que no puedes validar entregas si el sistema cree que tu almacén está vacío. Solucionar estos detalles sobre la marcha fue clave para entender cómo Odoo coordina los procesos y cómo, una vez ajustado, todo cobra sentido.

# 01

ent

orno y requisitos

Para arrancar este proyecto, tocaba adecuar el entorno para que Odoo Community se sintiera como en casa. Utilizamos un equipo con Windows 11, aunque, siendo sincero, para lo que nos ocupa, es casi como tener un Windows Server 2022. La arquitectura es la misma, las compatibilidades, también, y ambos soportan lo que Odoo necesita.

**1.1. Echando un vistazo al hardware y al sistema**

Antes de ponernos manos a la obra, repasamos los requisitos que Odoo recomienda. Según la documentación, necesitábamos al menos:

Una CPU de 2 núcleos (aunque con 4, la cosa va más fluida, sobre todo si hay varios usuarios).

4 GB de RAM, pero 8 GB son lo ideal para no tener problemas.

Unos 5 GB libres para la instalación, aunque yo reservaría unos 20 GB para bases de datos, módulos y demás archivos.

Un sistema de archivos NTFS, porque Odoo necesita permisos específicos.

Nuestro equipo cumplía con creces:

Intel Core i7

16 GB de RAM

SSD NVMe de 512 GB

NTFS

Windows 11 Pro

Con este hardware, no tuvimos ningún problema durante las pruebas, la verdad.

**1.2. Pequeños ajustes antes de empezar**

Antes de darle al instalador, hicimos algunos preparativos para evitar sorpresas:

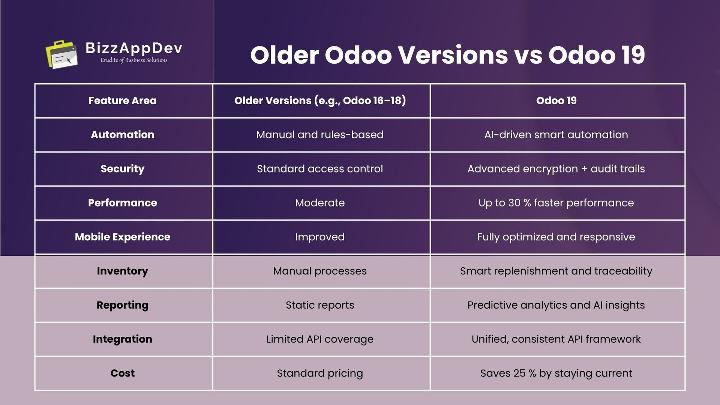
Actualizamos Windows a tope.

Desactivamos el antivirus mientras instalábamos.

Creamos una carpeta en C:\Odoo\_Instalacion\.

Activamos algunos componentes opcionales de Windows relacionados con .NET.

Con estos pequeños ajustes, dejamos el entorno listo para trabajar sin problemas.



# 02

Descargando Odoo Community

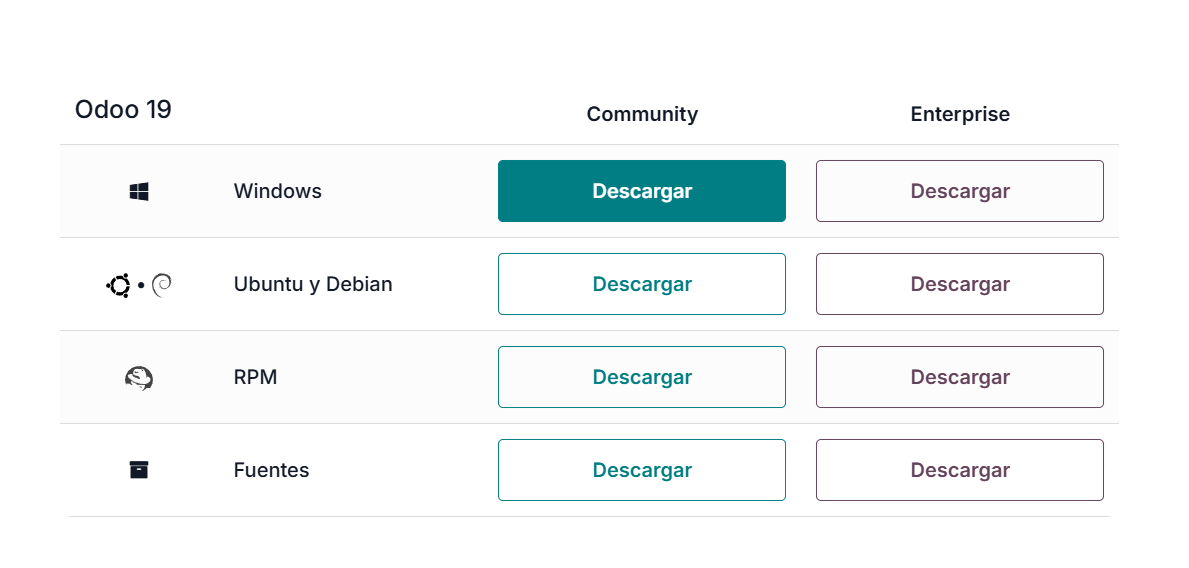
El siguiente paso era conseguir el instalador. Fuimos a la página de descargas de Odoo y elegimos:

Versión: Odoo 19.0 Community

Formato: Instalador para Windows (.exe)

Arquitectura: 64 bits

Guardamos el archivo en la carpeta que habíamos creado. Antes de ejecutarlo, verificamos el tamaño y la firma digital, algo que siempre recomiendo para evitar instalar software corrupto o manipulado.

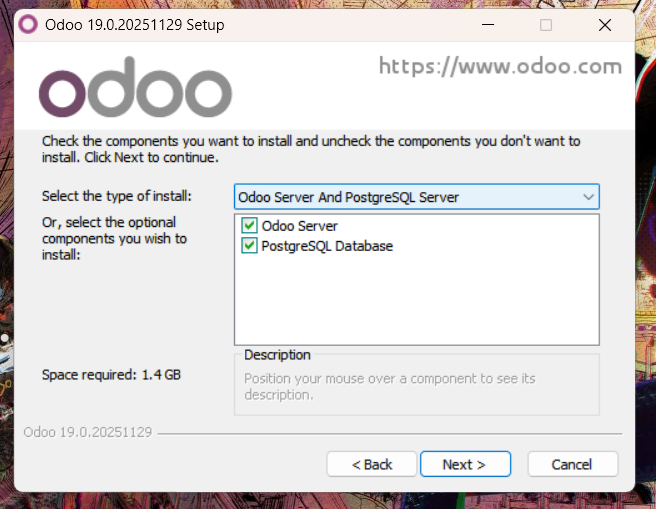


# 03

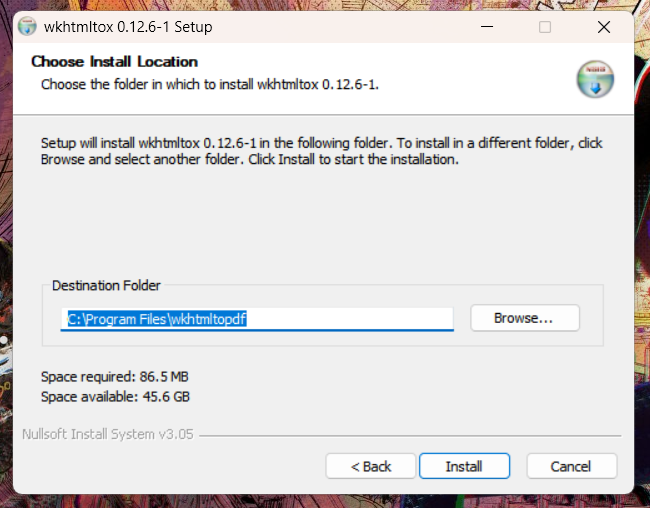
Instalando PostgreSQL

El instalador de Odoo para Windows es bastante completo. Incluye:

PostgreSQL (normalmente la versión 15) y Python con todas las librerías necesarias



Descargamos Wkhtmltopdf aparte (imprescindible para generar PDFs)



Durante la instalación, nos muestra las rutas donde se instalarán los componentes:

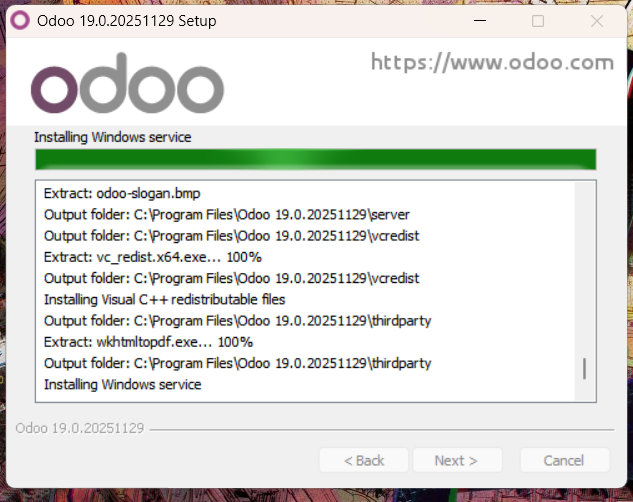
C:\Program Files\Odoo 19.0\

C:\Program Files\PostgreSQL\15\

También instala dos servicios:

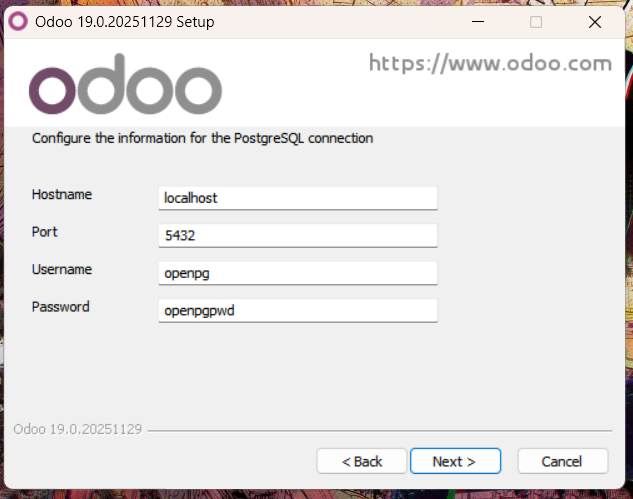
Servicio de Odoo (en modo automático)

Servicio de PostgreSQL (independiente)





En este punto, asignamos una contraseña segura para el usuario postgres, verificamos que los servicios se crean correctamente y comprobamos que el puerto por defecto (5432) esté operativo. Por último, revisamos en el Administrador de Tareas que PostgreSQL esté activo.



# 04

Instalando Odoo y creando la base de datos inicial

Una vez terminada la instalación, entramos en Odoo desde el navegador:

http://localhost:8069

La primera vez que carga, aparece un asistente para crear la base de datos. Configuramos lo siguiente:

Nombre: erp\_pymes2025

Correo del administrador

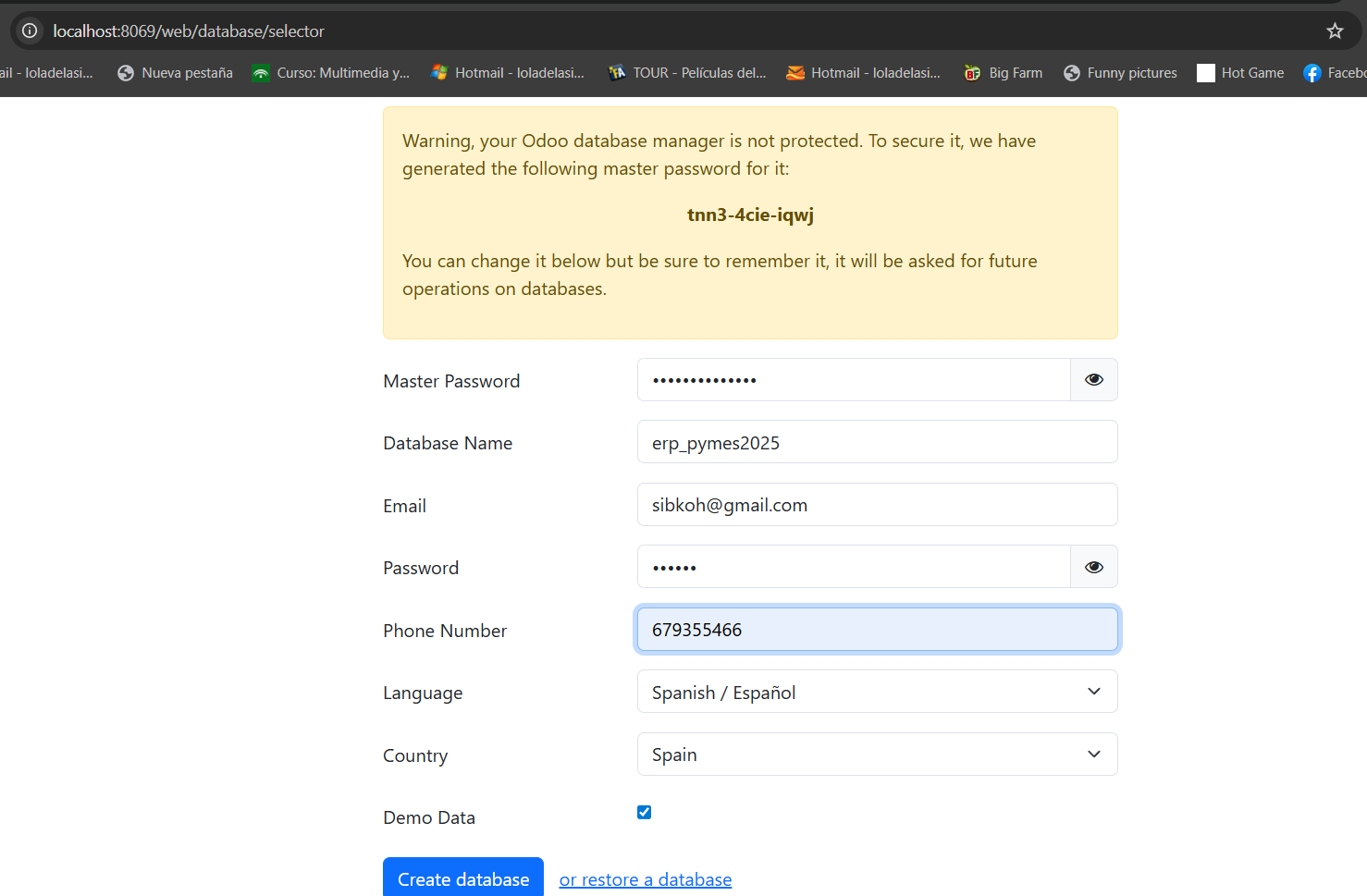
Contraseña del administrador

Idioma: Español

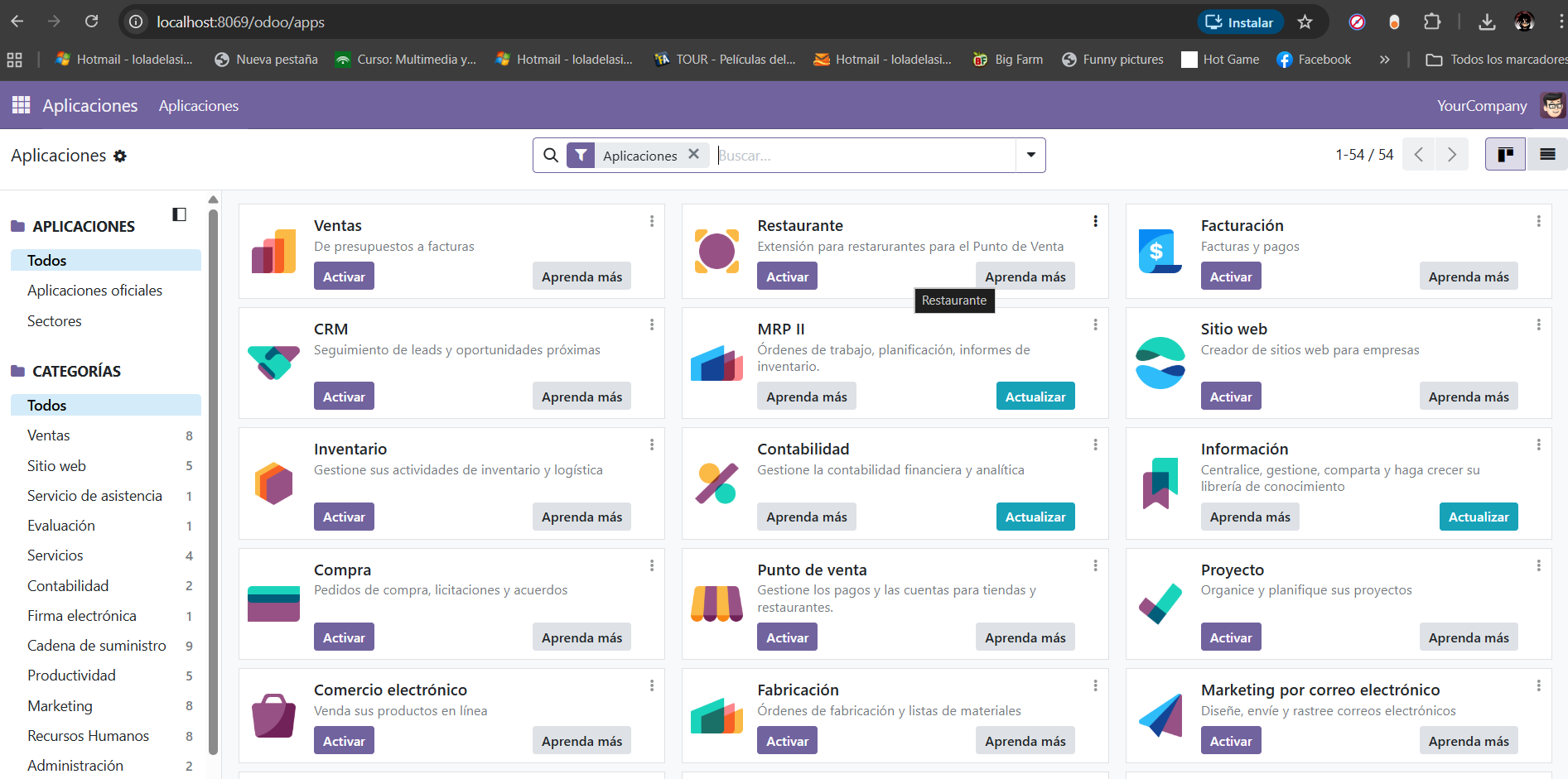
País: España

Moneda: EUR

Demo data activado (para tener algunos ejemplos)



Después de unos minutos, Odoo genera la base de datos y nos lleva directamente al panel principal.



# 05

Configuración básica del sistema

**5.1. Ajustes generales**

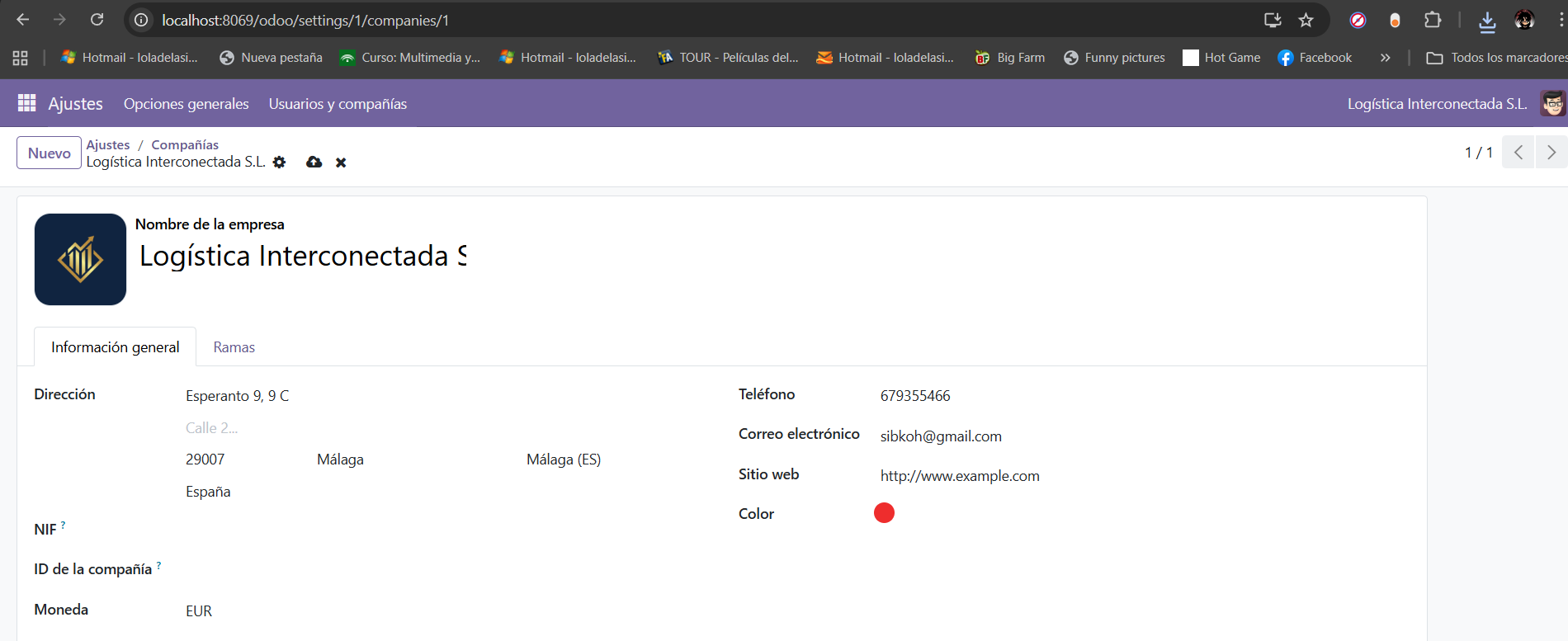
Desde el menú Ajustes, completamos la información básica de la empresa:

Nombre: Logística Interconectada S.L.

Dirección fiscal

Logo

Zona horaria: Europe/Madrid



**5.2. Idioma y moneda**

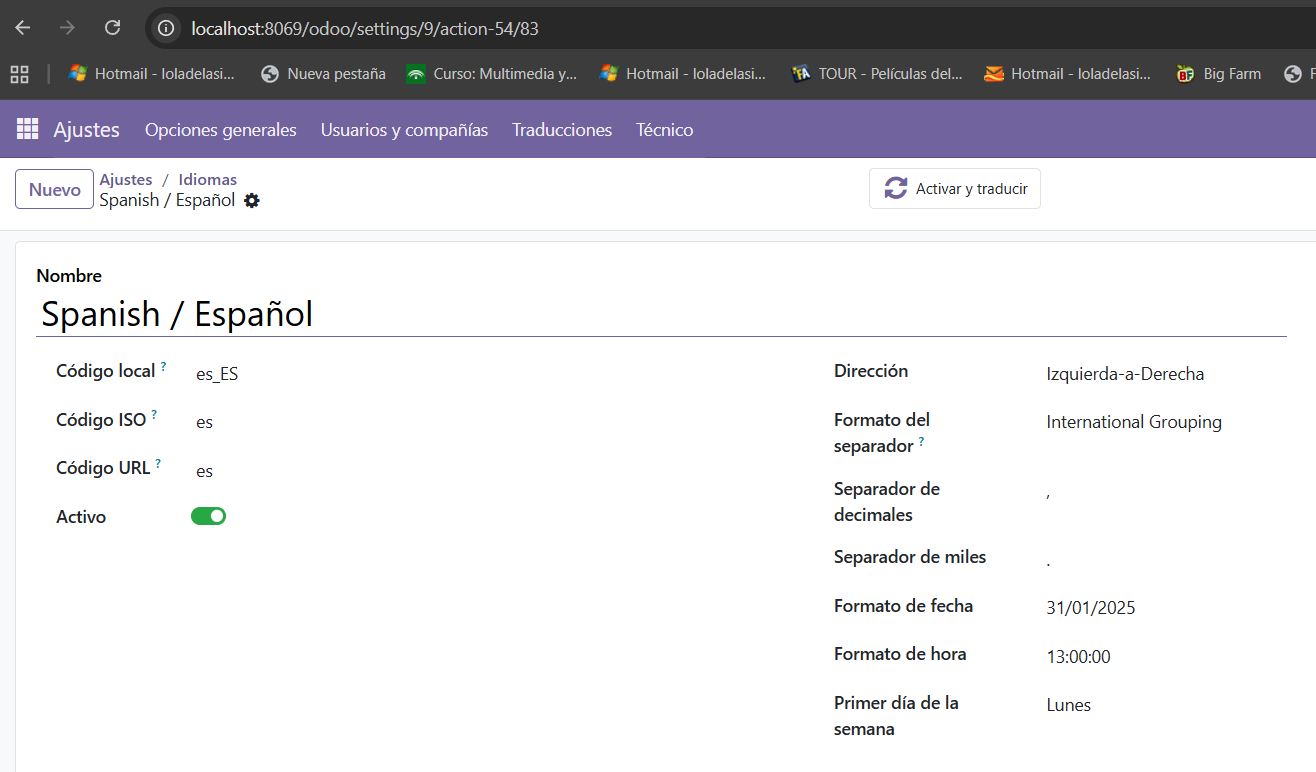
Estos ajustes ya los habíamos seleccionado antes, pero los revisamos de nuevo:

Formato de fecha

Separador decimal

Moneda principal

Pequeños detalles que pueden evitar errores contables más adelante, en mi opinión.



# 06

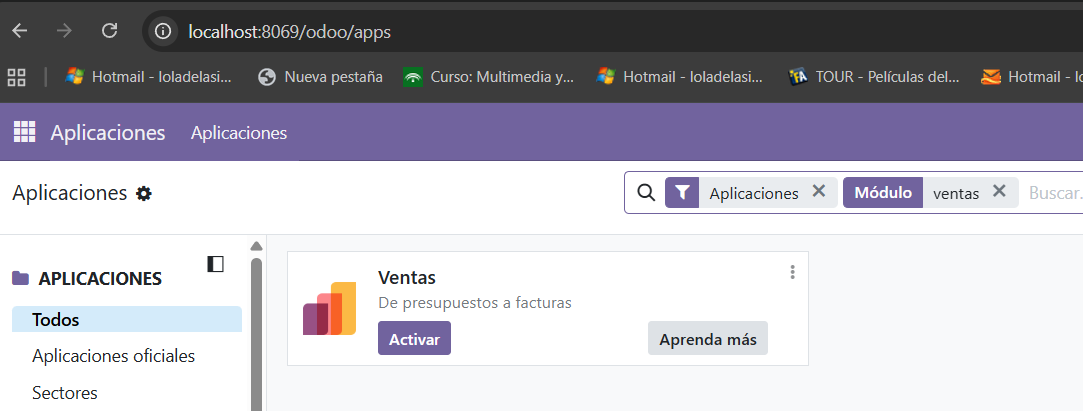
Creación de usuarios y asignación de roles

Bien, vayamos al grano con esta prueba de Odoo para una pyme:

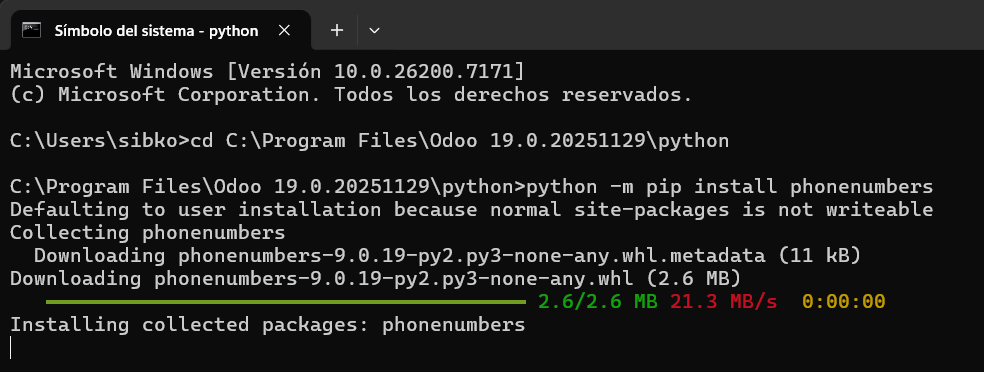
Primero, lo esencial: **Ventas** e **Inventario**. Los instalamos desde Ajustes, en la sección de Aplicaciones.

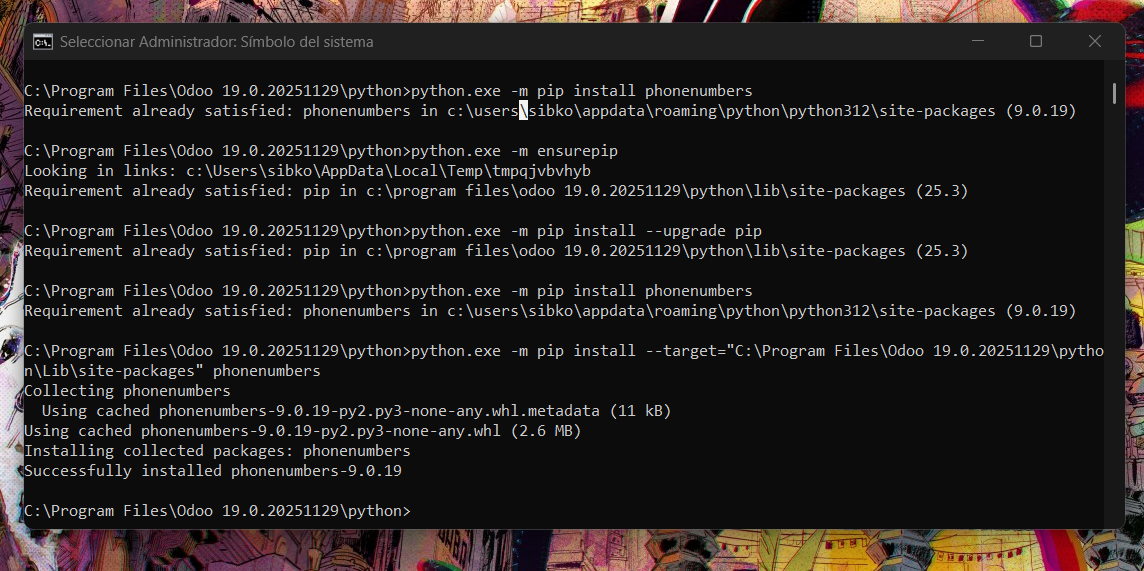
El **módulo de Ventas** nos da herramientas para presupuestos, gestionar clientes y pasar de presupuesto a pedido.

El de **Inventario**, control de stock, entradas, salidas, etc.

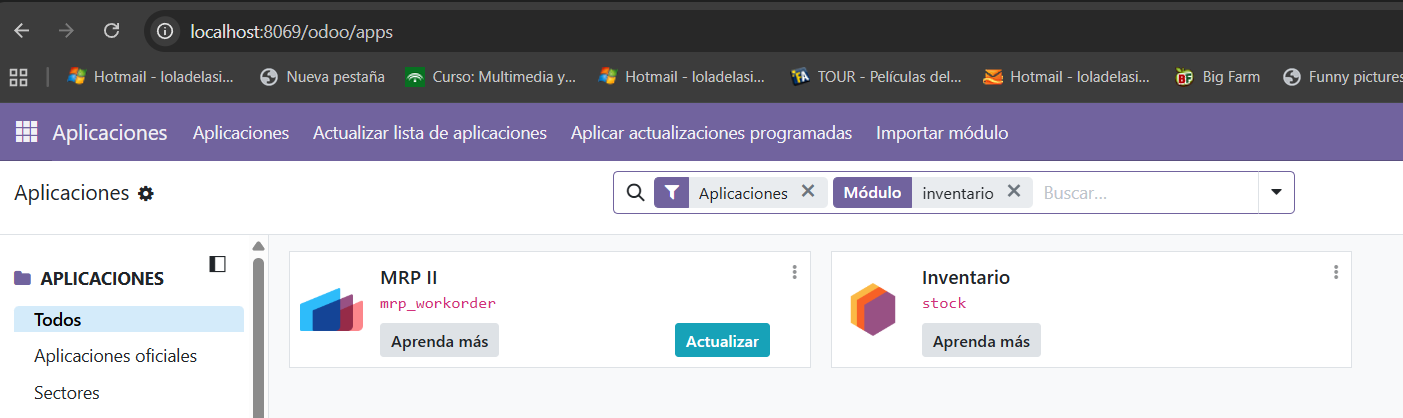


Para poder activar el módulo ventas e inventario, tenemos que instalar una dependencia de Python que no instaló Odoo originalmente.





Cerramos Odoo y reiniciamos los servicios de Odoo y Postgre.

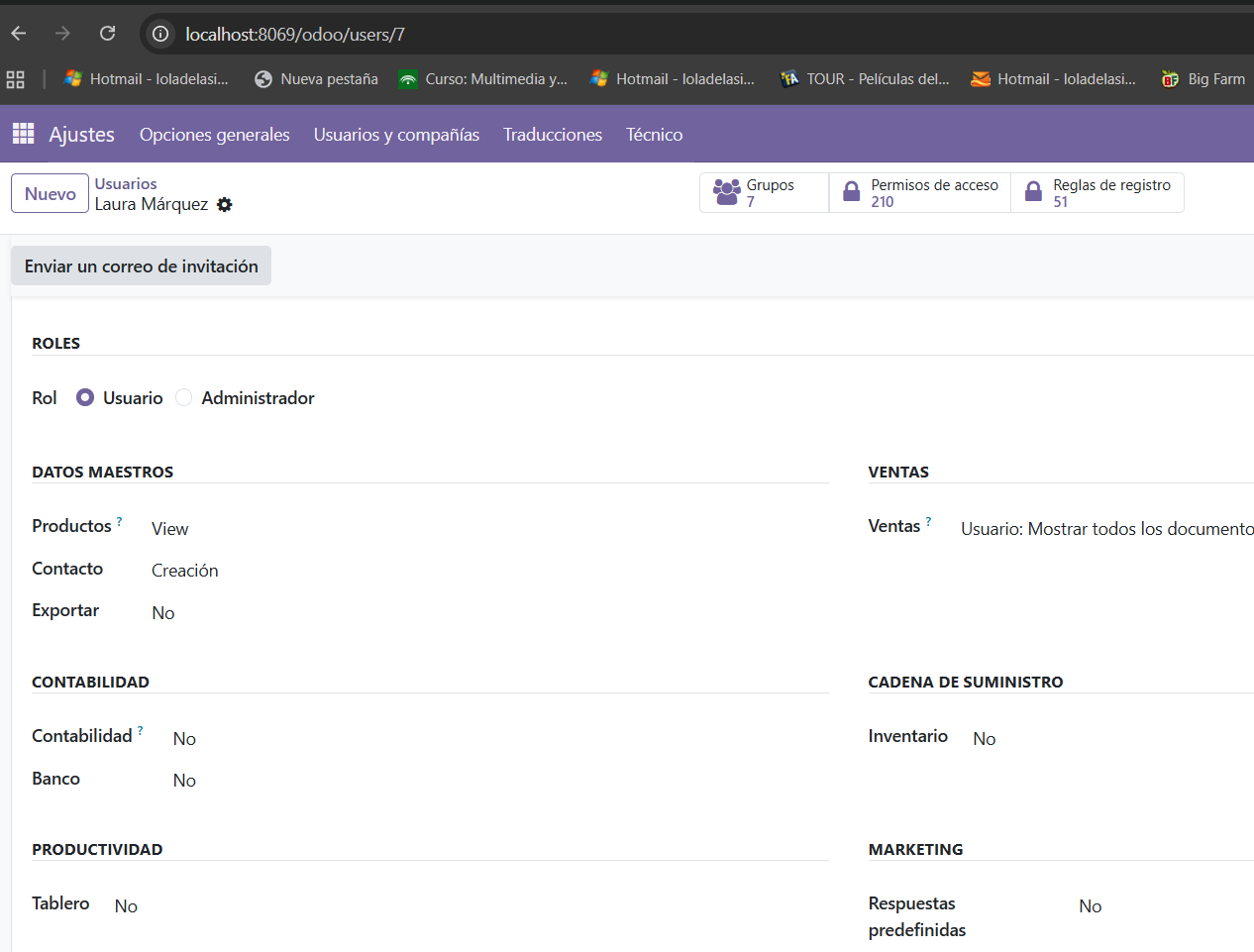


Vemos que ya las hemos podido activar, y así poder trabajar con esos modulos.

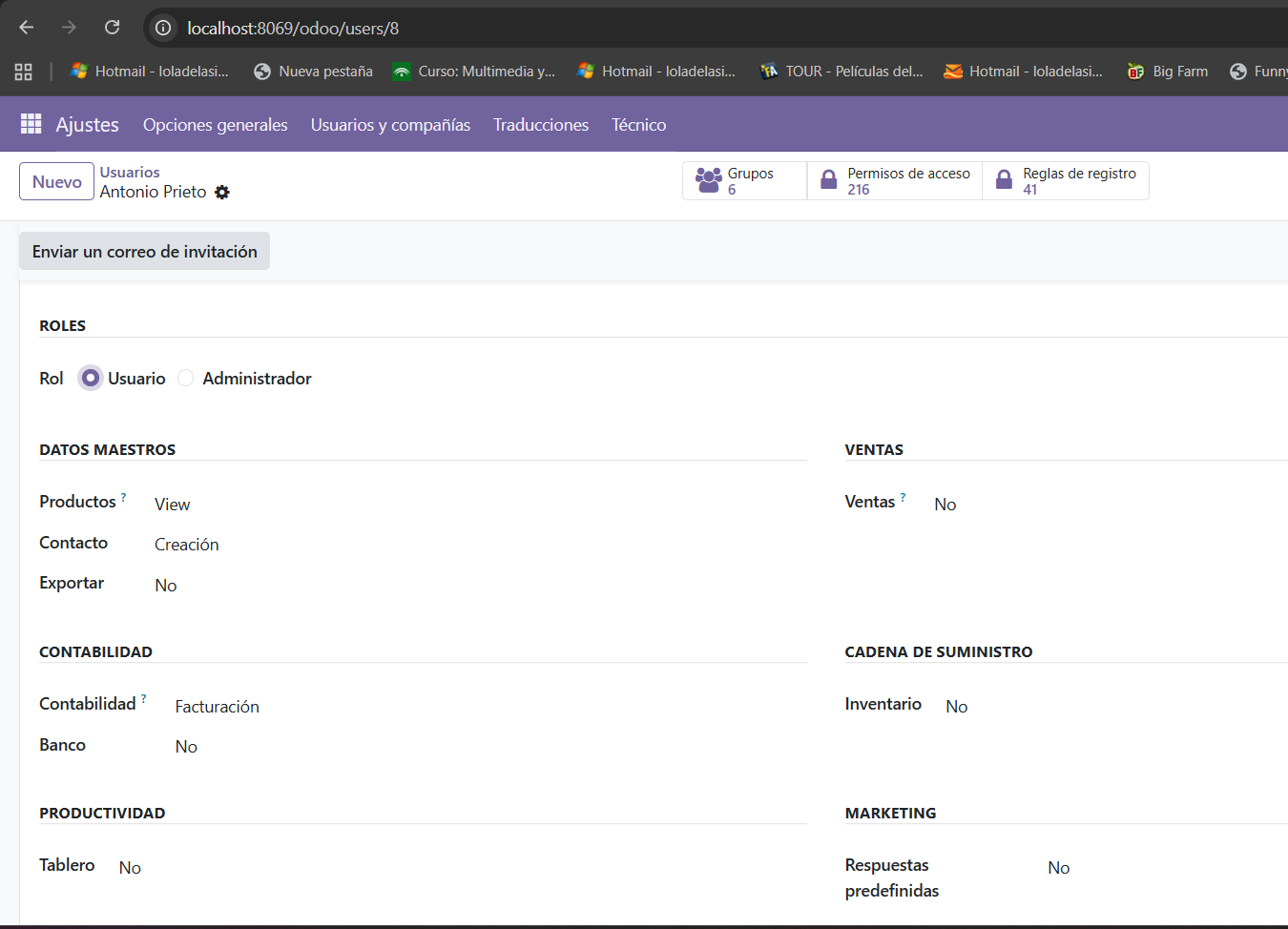
Creamos tres **usuarios** para simular una estructura.

Creamos unos usuarios básicos con sus roles.

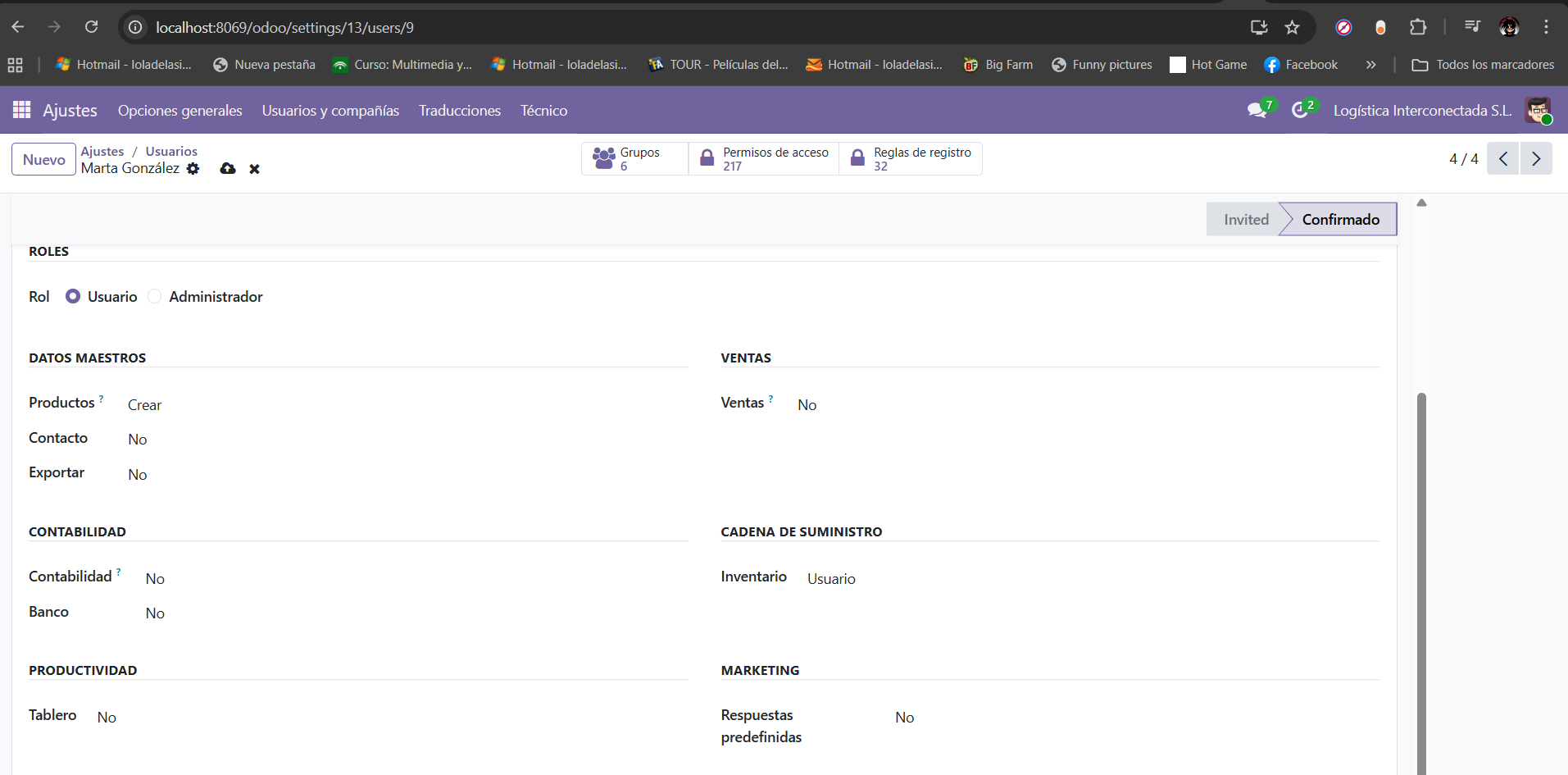
Laura Márquez, la encargada de ventas, tiene acceso total al módulo de Ventas, CRM y Contactos.



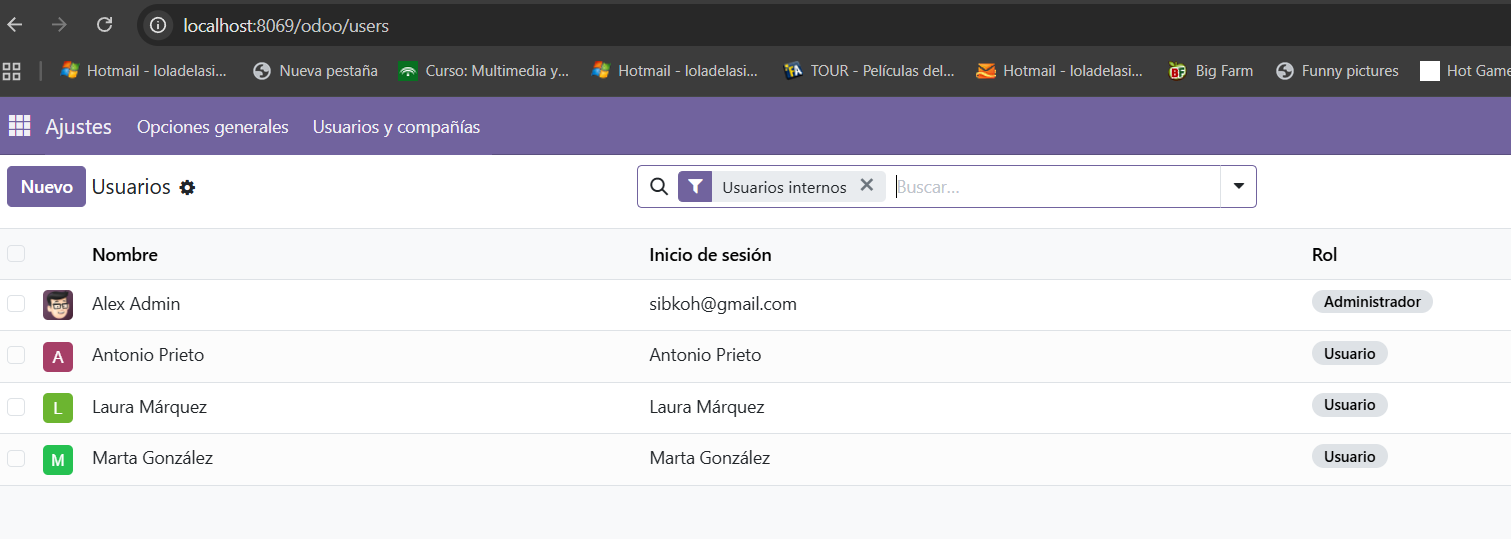
Antonio Prieto, que lleva la contabilidad, pues a Contabilidad y Facturación.

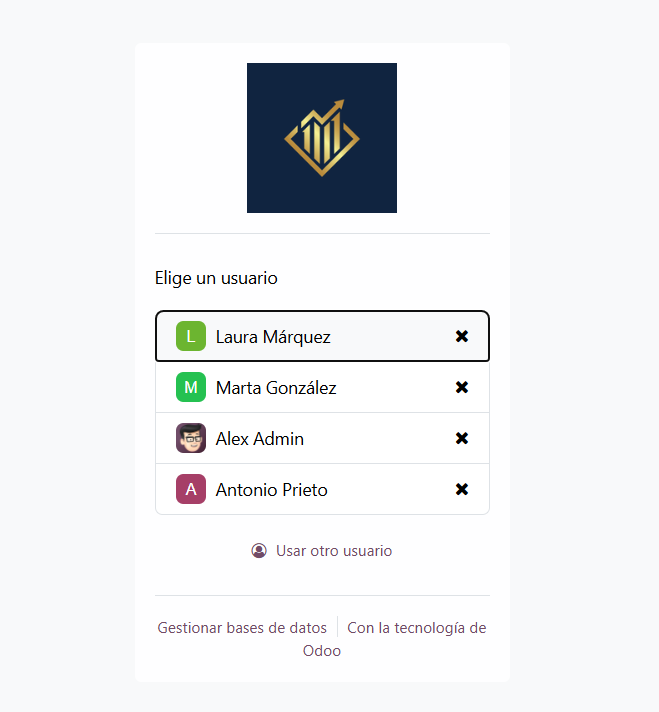


Y Marta González, en el almacén, lógicamente a Inventario y Logística.



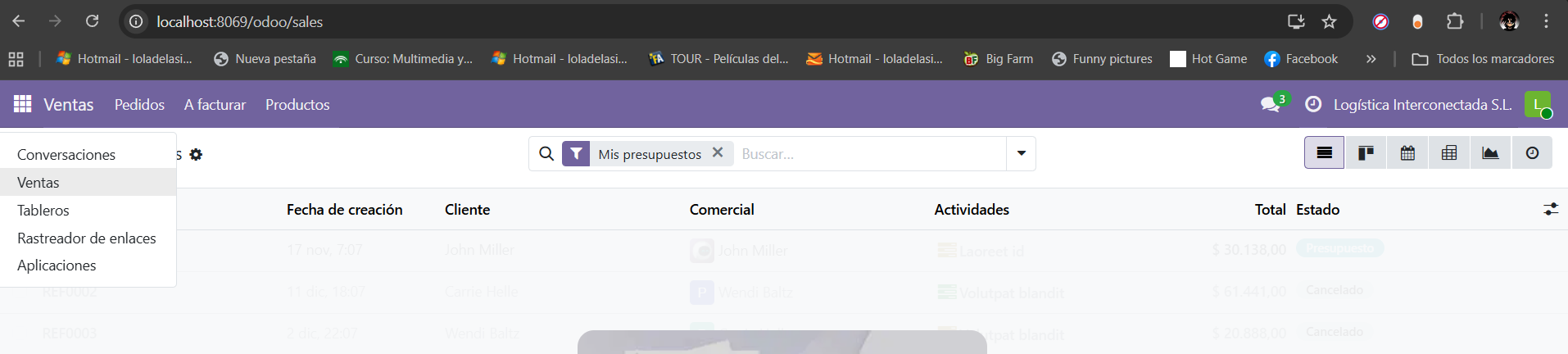
Verificamos que cada uno ve solo lo que necesita, y ahí vemos cómo Odoo se adapta a cada rol.



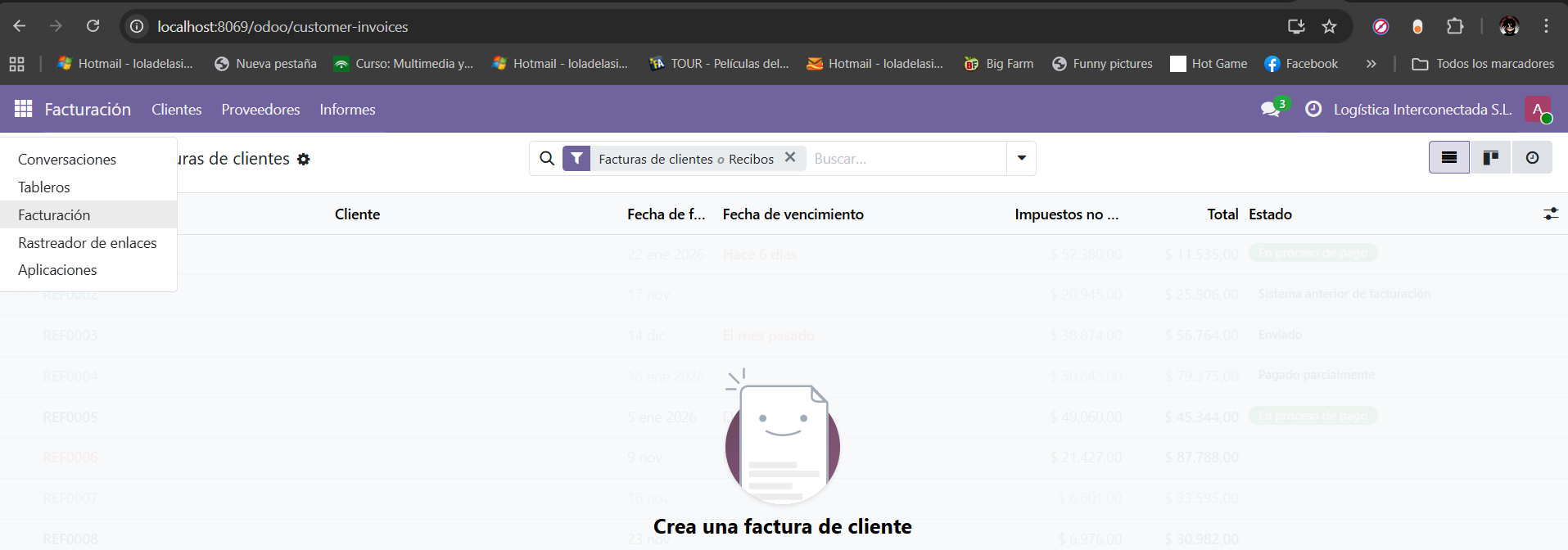


Cerramos sesión de nuestro usuario administrador y accedemos uno por uno a cada uno de los usuarios creados con su login y contraseña, para ver que todo está correcto y accede a los módulos permitidos.

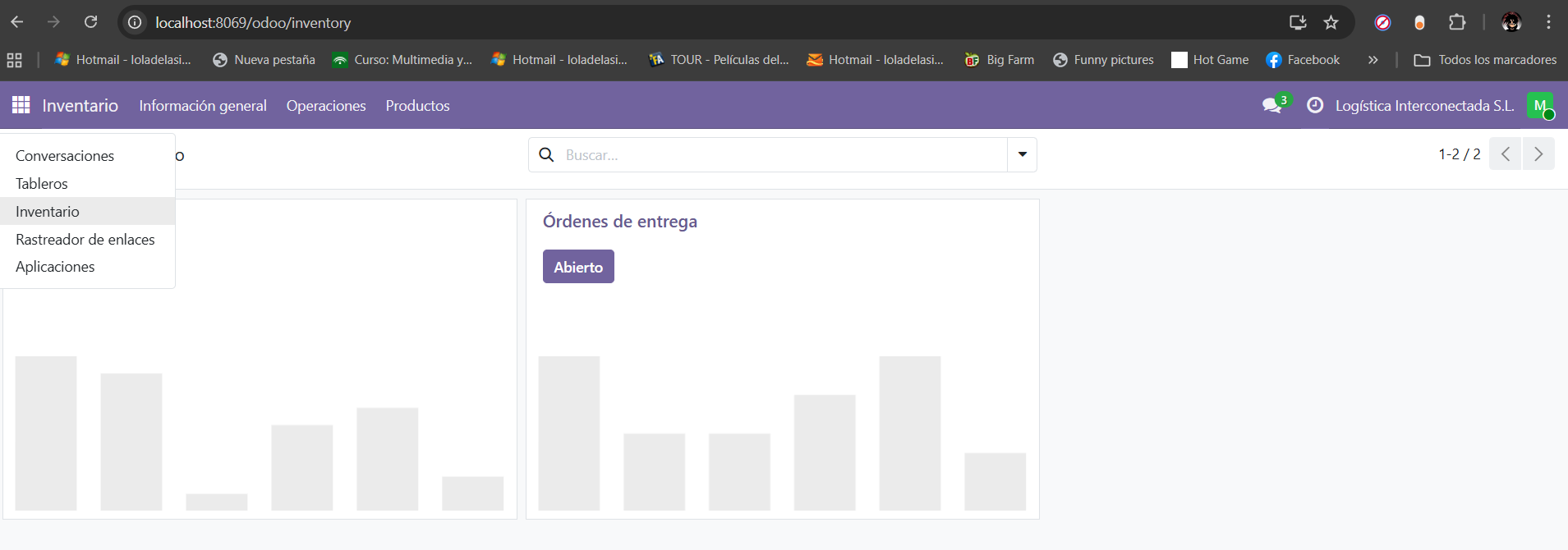
Laura



Antonio



Marta



Todo va correcto hasta ahora.

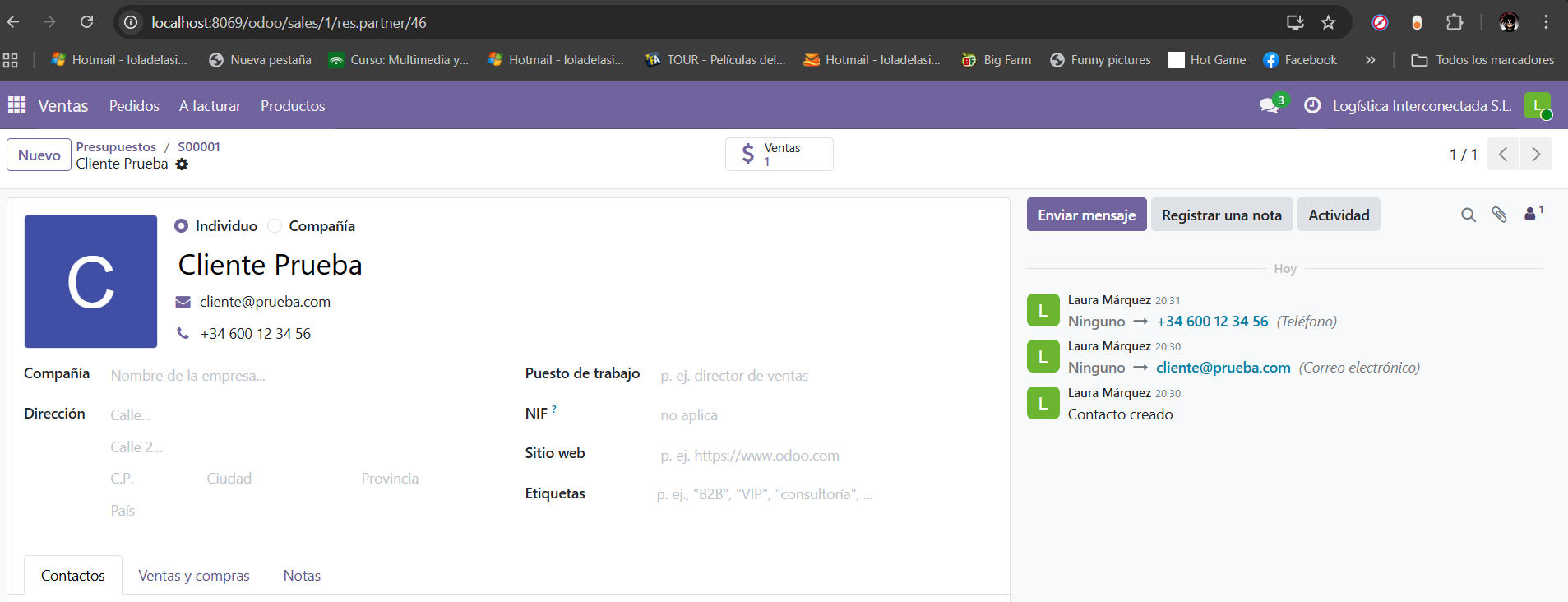
# 07

prueba de interconexión de modulos

Para estar seguros de que el ERP no solo funciona, sino que fluye, pusimos a prueba la comunicación real entre Ventas e Inventario. No queríamos quedarnos en la teoría, así que realizamos el recorrido completo, desde el momento en que damos de alta a un cliente hasta que confirmamos que el stock se ha movido. Aquí te cuento exactamente cómo lo hicimos.

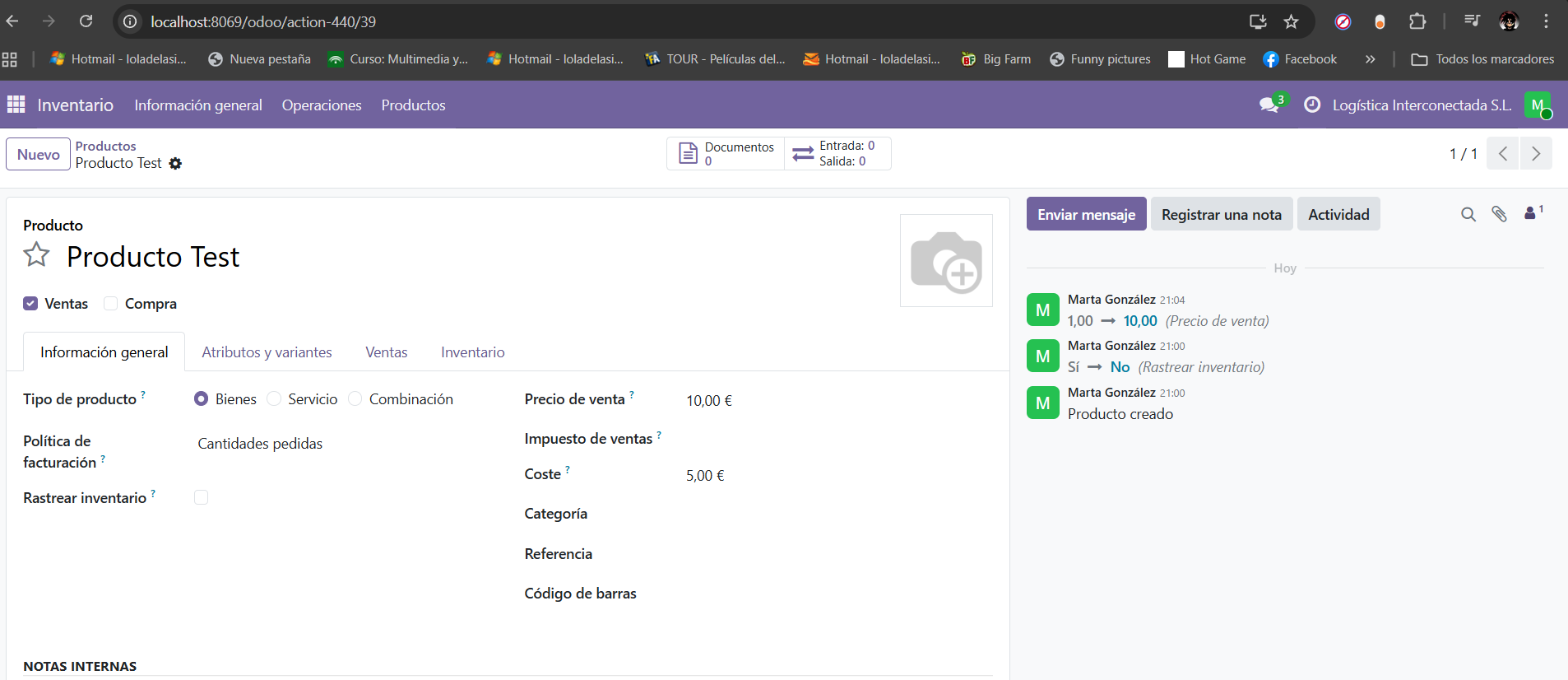
**Paso 1: Crear al cliente de prueba**

Todo empieza con quién nos compra. Iniciamos sesión con un usuario de Ventas (Laura) y fuimos directo a crear un cliente nuevo. No nos complicamos demasiado, simplemente introdujimos los datos necesarios para que el sistema tuviera a alguien real a quien facturar: le pusimos de nombre Cliente Prueba, un correo ficticio y un teléfono. Guardamos la información y listo. El objetivo aquí era simple: tener una ficha válida que pudiéramos usar durante todo el experimento.

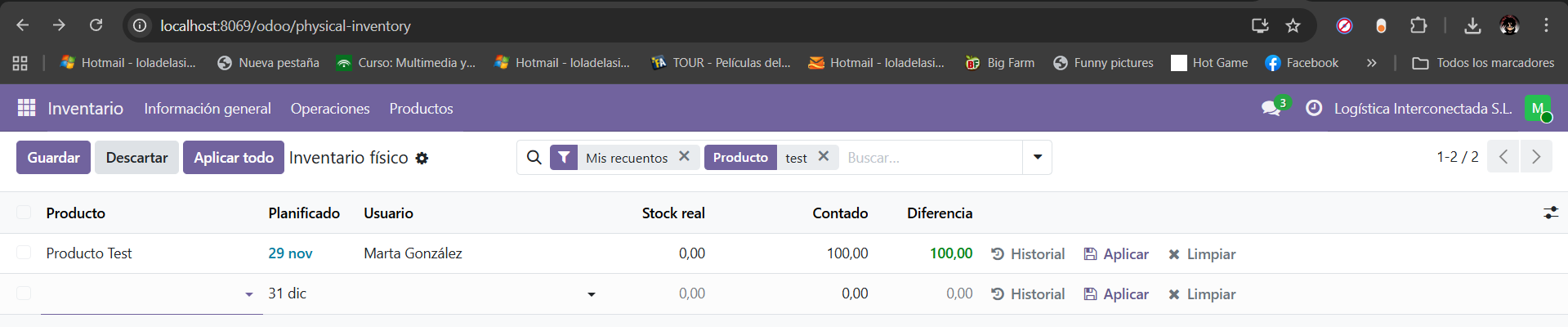


**Paso 2: Generar el pedido de venta**

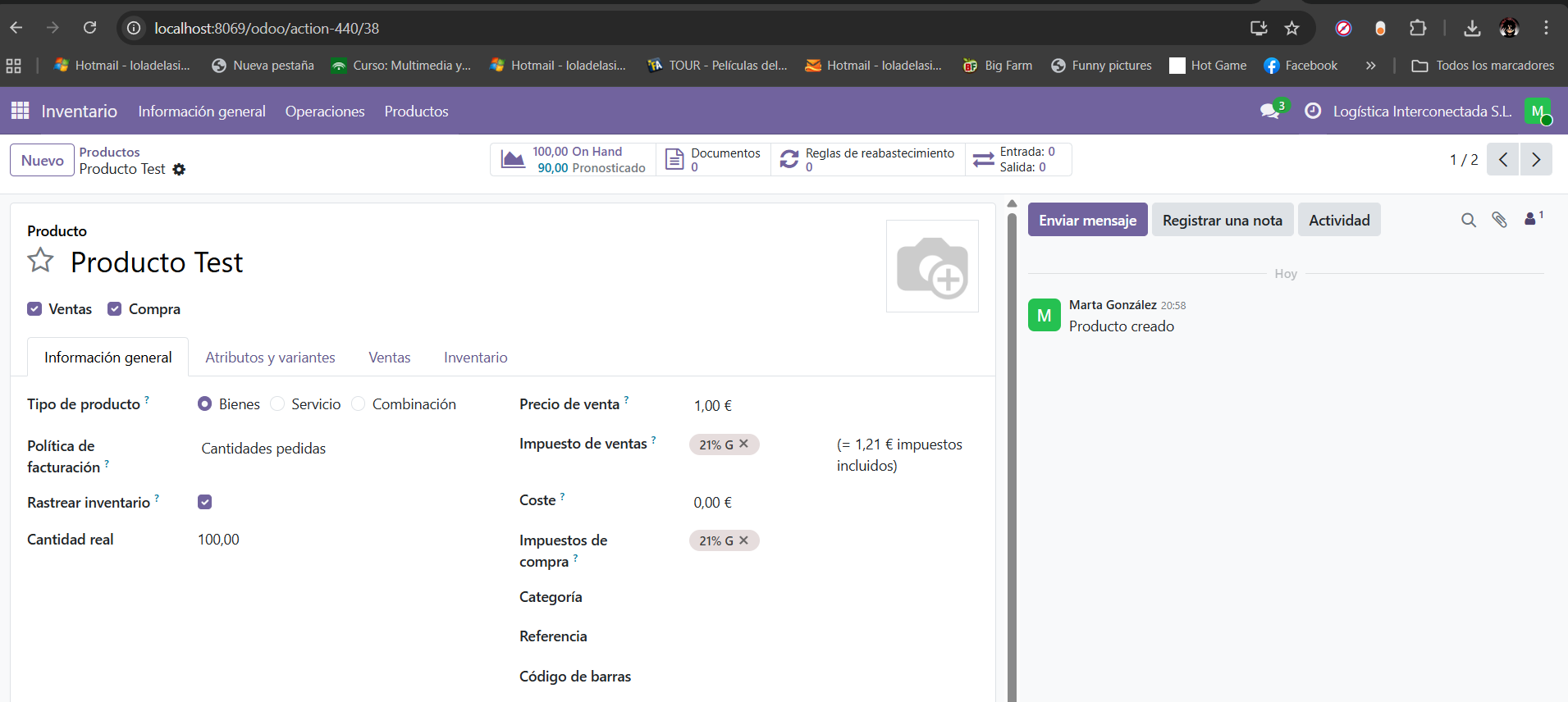
Creamos un producto en Inventario



Creamos stockage del producto



Le damos a aplicar y vemos que ya disponemos de 100 stocks de nuestro producto creado.

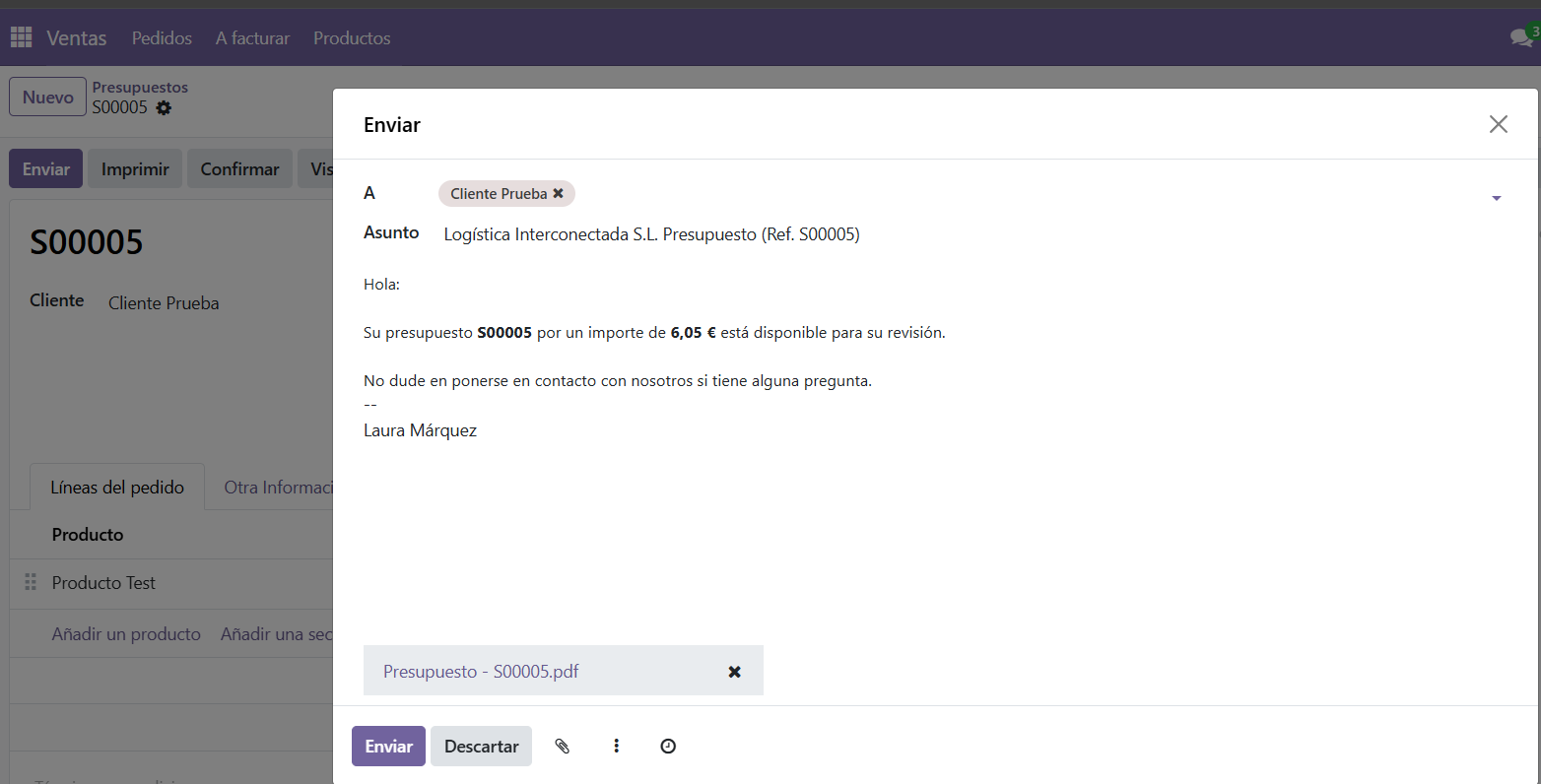


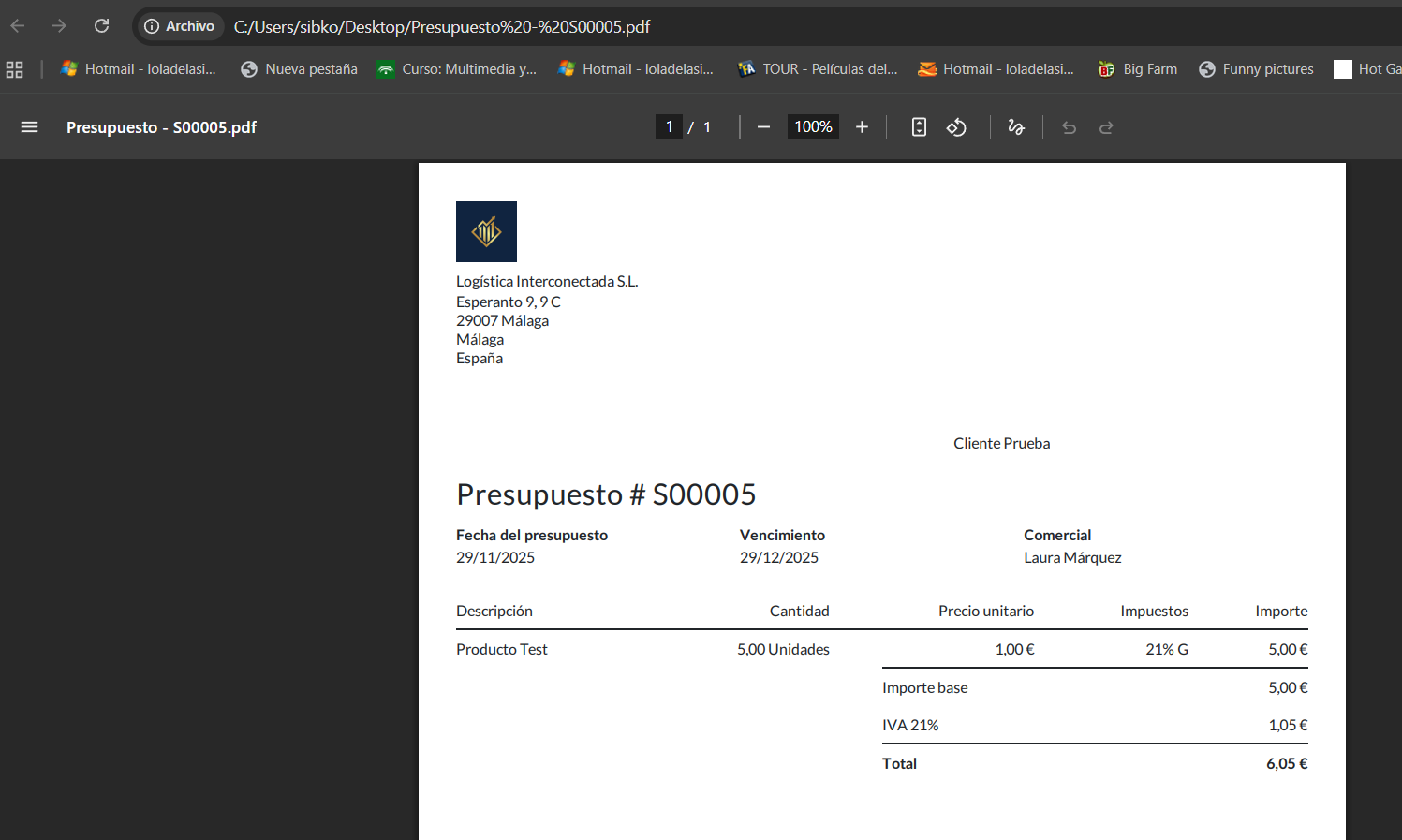
Con el cliente ya en el sistema, nos fuimos a la sección de Pedidos de venta. Seleccionamos a nuestro nuevo Cliente Prueba y añadimos el producto que ya hemos creado y tenía existencias en el almacén, el Producto Test, y le asignamos unas 5 unidades. Echamos un vistazo rápido para asegurarnos de que el precio y los impuestos tuvieran sentido y guardamos el pedido. Este paso es clave porque es el detonante que debería despertar al módulo de Inventario.

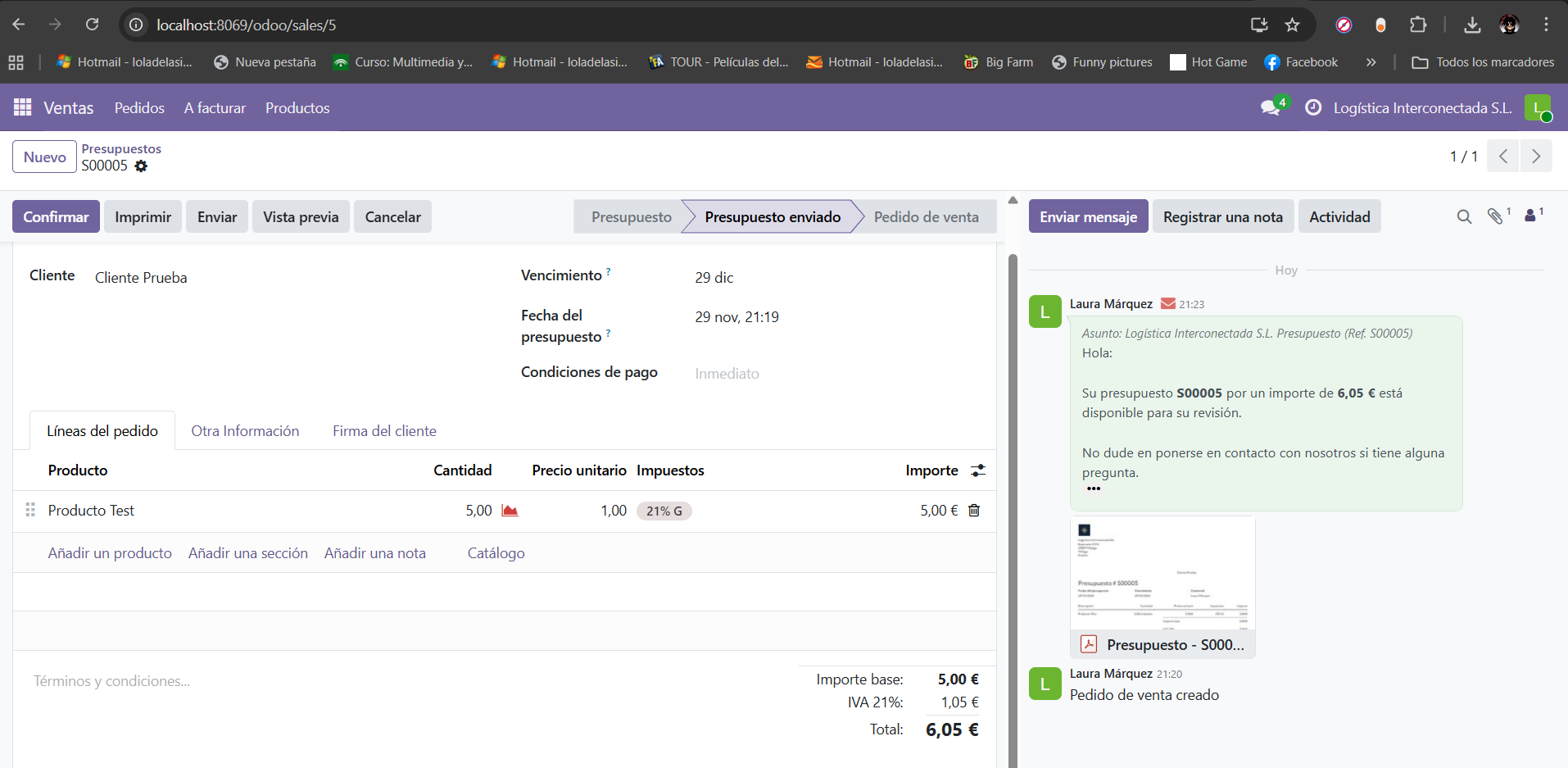


**Paso 3: Confirmar la operación**

Con el pedido abierto, hicimos clic en Confirmar. Aquí es donde el sistema debería hacer su magia automáticamente: genera una orden de entrega para el equipo de almacén y actualiza el resumen en ventas. Si la configuración está bien hecha, esto ocurre de inmediato sin que salte ningún error ni advertencia extraña.



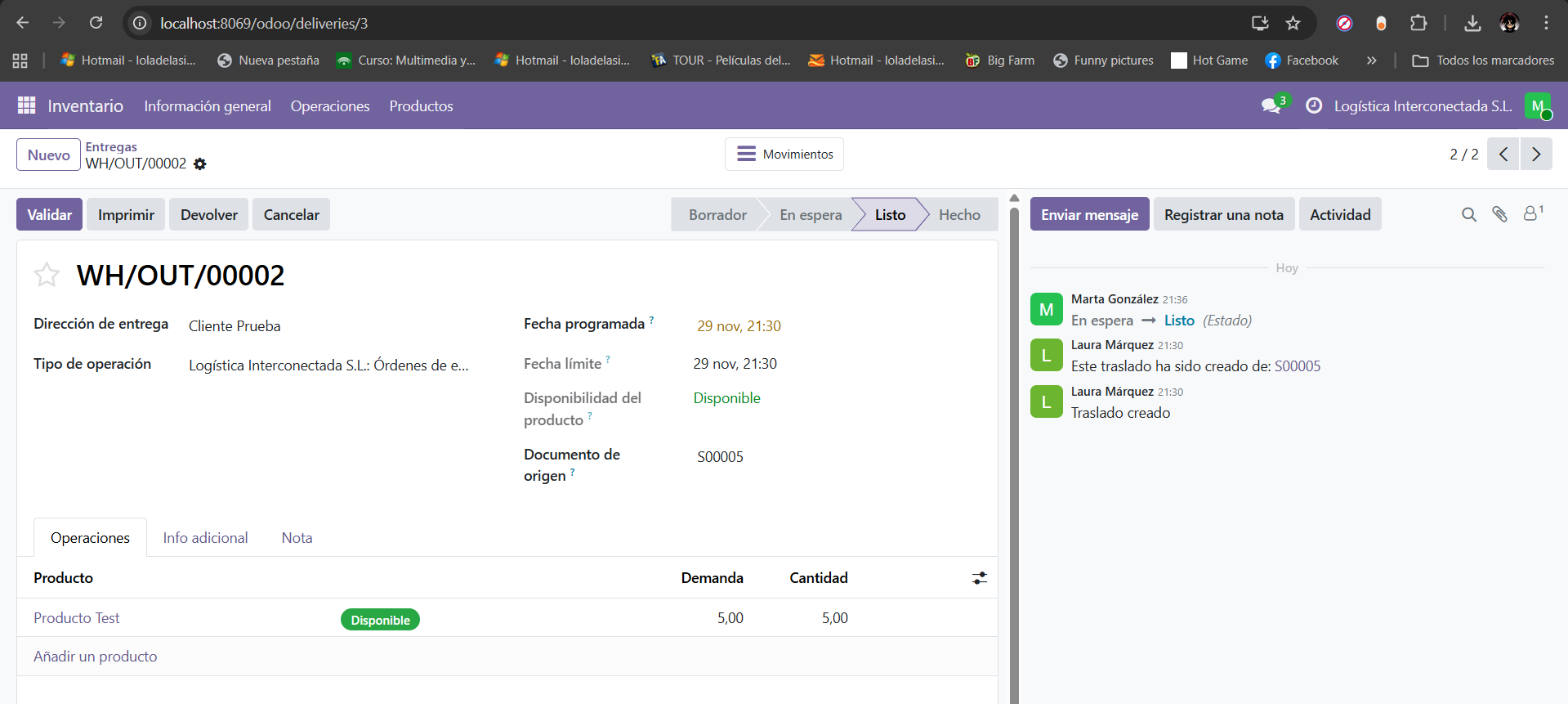




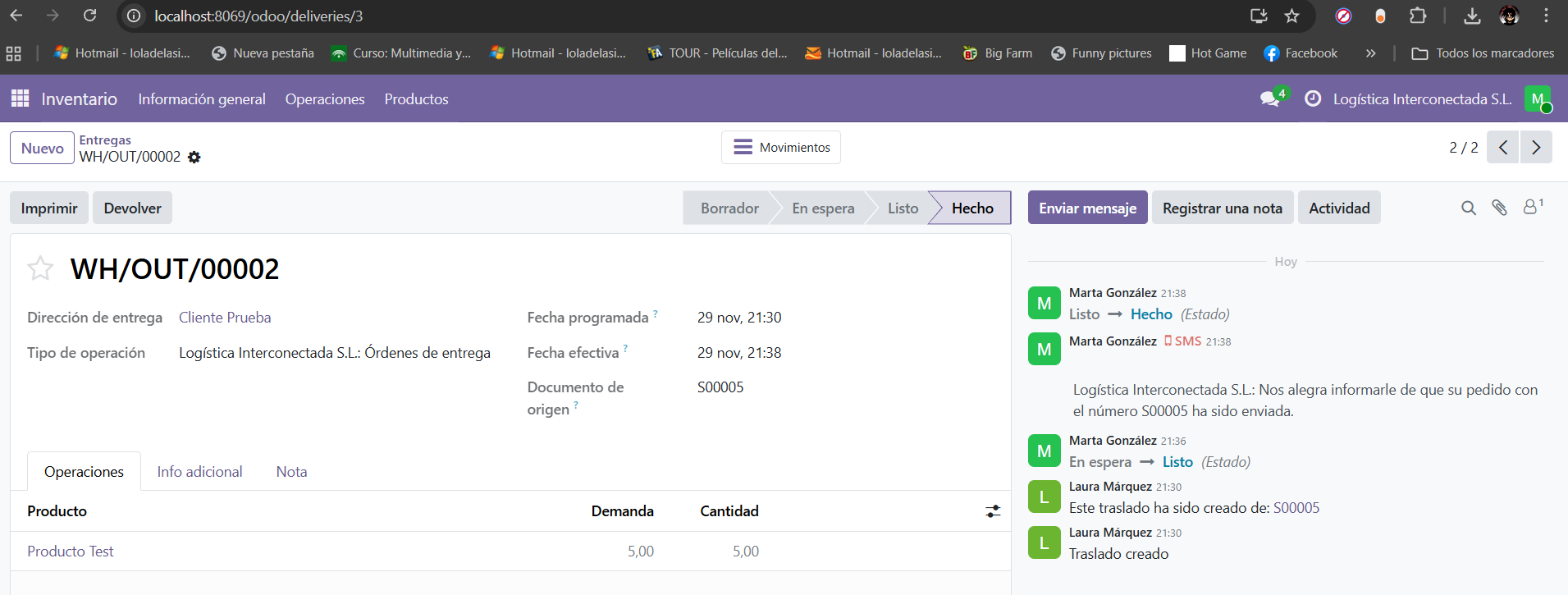
Confirmamos el pedido y listo para que esté en el sistema.

**Paso 4: Validar desde el almacén**

Para esta parte entramos con un usuario de Inventario para revisar las operaciones. Fuimos a las Transferencias y ahí estaba la orden de entrega que se había creado sola desde ventas. Comprobamos que coincidiera el cliente y que fueran las unidades correctas.



Al darle a Validar, el sistema descuenta esas existencias del inventario. Esto es vital porque nos confirma que las acciones de un departamento están impactando directamente en el otro.



**Paso 5: Verificar que el ciclo se cierra**

Para terminar, regresamos al módulo de Ventas y abrimos de nuevo el pedido original. Lo que buscábamos era ver que el estado hubiera cambiado a Completado o Entregado. Ver esa confirmación es la prueba definitiva de que el ERP coordinó los dos mundos perfectamente, actualizando las cantidades y cerrando el flujo de trabajo sin perder datos por el camino.



# 08

conclusiones

Al final, logramos que la instalación funcionara y que los módulos de Ventas e Inventario se hablaran entre sí tal como queríamos. Pero para llegar ahí tuvimos que resolver varios tropiezos que vale la pena mencionar:

Dependencias que faltaban en Odoo

El módulo de Ventas se quejaba porque le faltaba la librería phonenumbers.

Lo arreglamos instalándola manualmente con pip dentro del directorio de Odoo y listo.

El sistema no veía Wkhtmltopdf

Al principio era imposible generar un PDF porque Odoo no encontraba la herramienta.

Tuvimos que añadir la ruta de instalación a las variables de entorno del sistema y reiniciar. A partir de ahí, los documentos empezaron a salir perfectos.

Stock a cero

Intentábamos validar entregas y el sistema nos frenaba porque no había mercancía.

La solución fue lógica: ajustamos el inventario inicial con cantidades de prueba para que el flujo pudiera continuar.

El problema del usuario demo

El usuario de demostración nos estaba generando ruido en los datos.

Decidimos trabajar con usuarios reales y permisos limpios para evitar errores heredados.

Propuestas de mejora

Viendo cómo fue todo, hay un par de cosas que harían que la próxima vez sea mucho más fluida:

Documentación clara: Haría falta un manual sencillo, de esos que cualquiera entiende, para enseñar a crear productos y ajustar stock sin miedo a romper nada.

Automatizar el inicio: Tener plantillas de productos con stock ya cargado nos ahorraría mucho tiempo en las pruebas iniciales.

Protocolo de pruebas: Antes de decir que algo está listo, deberíamos tener una lista de chequeo obligatoria para validar que todos los módulos responden bien.

Formación real: No basta con dar acceso; hay que enseñar al equipo a confirmar pedidos y gestionar los informes para que le saquen provecho a la herramienta.

Resumen final

Esta experiencia nos ha servido para confirmar que Odoo Community, cuando se configura con cariño, es una bestia para la gestión integral. Hemos aprendido a sortear los baches técnicos habituales y tenemos claro qué mejorar para el futuro. Al final del día, lo que queda demostrado es que un buen ERP es la diferencia entre ir apagando fuegos o tener una operativa logística eficiente y profesional.

# 09

referencias

<https://www.odoo.com/documentation/19.0/es/applications.html>

[https://www.techstartingpoint.com/es/instalar-y-usar-odoo-en-windows/](https://www.techstartingpoint.com/es/instalar-y-usar-odoo-en-windows/?utm_source=chatgpt.com)

[https://www.odoo.com/es/forum/ayuda-1/wkhtmltopdf-could-not-be-found-on-this-system-286825](https://www.odoo.com/es/forum/ayuda-1/wkhtmltopdf-could-not-be-found-on-this-system-286825?utm_source=chatgpt.com)

[https://www.odoo.com/documentation/13.0/es/administration/install/install.html#](https://www.odoo.com/documentation/13.0/es/administration/install/install.html?utm_source=chatgpt.com#)

[https://www.odoo.com/documentation/19.0/es/](https://www.odoo.com/documentation/19.0/es/?utm_source=chatgpt.com)

[https://www.odoo.com/documentation/19.0/es\_419/applications/inventory\_and\_mrp/inventory.html](https://www.odoo.com/documentation/19.0/es_419/applications/inventory_and_mrp/inventory.html?utm_source=chatgpt.com)

[https://www.odoo.com/documentation/19.0/es\_419/applications/studio/pdf\_reports.html](https://www.odoo.com/documentation/19.0/es_419/applications/studio/pdf_reports.html?utm_source=chatgpt.com)