

PRÁCTICA: Proyecto Final de Implantación y Trazabilidad (Hardware Solutions S.L.)

Módulo: Sistemas de Gestión Empresarial (UT4, UT5 y UT6)

Nivel: 2º DAM.

Ignacio Gómez Trigueros 2º DAM

Hito 1

Código:

```
import xmlrpc.client
import csv

# Configuración de conexión
url = "https://sgempresarial1.odoo.com"
db = "sgempresarial1"
username = "896846@alu.murciaeduca.es"
password = "ArchenaIgnacio2025"

# ----- CONEXIÓN -----
common = xmlrpc.client.ServerProxy(f"{url}/xmlrpc/2/common")
uid = common.authenticate(db, username, password, {})

if uid:
    print("Conexión exitosa. UID:", uid)
else:
    print("Error en la autenticación.")

# ----- OPERACIONES -----
models = xmlrpc.client.ServerProxy(f"{url}/xmlrpc/2/object")

# Obtener productos
products = models.execute_kw(db, uid, password,
    'product.template', 'search_read',
    [[]], {'fields': ['name', 'list_price', 'standard_price']})

# Guardar en CSV
csv_file = 'catalogo_hardware.csv'

with open(csv_file, mode='w', newline='', encoding='utf-8') as file:
    writer = csv.writer(file)
    # Escribir encabezados
    writer.writerow(['Producto', 'Precio de venta', 'Coste'])
    # Escribir cada producto
    for product in products:
        writer.writerow([product['name'], product['list_price'],
            product['standard_price']])

print(f"Datos guardados en {csv_file}")
```

Se realizó una validación previa mediante la herramienta de Test de Importación de Odoo 17, obteniendo el mensaje "Todo parece correcto", lo que confirma la coherencia estructural y referencial del dataset antes de su carga definitiva en el sistema.

```
PS C:\Users\Josander\OneDrive\Escritorio\GestionFinal\ProyectoGestionFinal> & C:\Users\Josander\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Josander/OneDrive/Escritorio/GestionFinal/ProyectoGestionFinal/Hito1.1.py
Conexión exitosa. UID: 2
Producto: Booking Fees, Precio: 50.0, Coste: 0.0
Producto: DUA VAT Valuation 10%, Precio: 1.0, Coste: 0.0
Producto: DUA VAT Valuation 21%, Precio: 1.0, Coste: 0.0
Producto: DUA VAT Valuation 4%, Precio: 1.0, Coste: 0.0
Producto: Figura Orco, Precio: 50.0, Coste: 0.0
Producto: Figura Pikachu, Precio: 25.0, Coste: 0.0
Producto: Procesador Intel Core i7-13700K, Precio: 420.0, Coste: 350.0
Producto: tornillos, Precio: 1.0, Coste: 0.0
PS C:\Users\Josander\OneDrive\Escritorio\GestionFinal\ProyectoGestionFinal>
```

```
PS C:\Users\Josander\OneDrive\Escritorio\GestionFinal\ProyectoGestionFinal> & C:\Users\Josander\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Josander/OneDrive/Escritorio/GestionFinal/ProyectoGestionFinal/Hito_1.py
Archivos generados correctamente.
PS C:\Users\Josander\OneDrive\Escritorio\GestionFinal\ProyectoGestionFinal>
```

```
Conexión exitosa. UID: 2
Datos guardados en catalogo_hardware.csv
PS C:\Users\Josander\OneDrive\Escritorio\GestionFinal\ProyectoGestionFinal>
```

Queda así mi inventario en Odoo:

☆ **Figura Orco**

Precio: 50,00 €

Stock real: 20,00

☆ **Figura Pikachu**

Precio: 25,00 €

Stock real: 100,00

☆ **Procesador Intel Core i7-13700K**

[HW001]

Precio: 420,00 €

☆ **tornillos**

Precio: 1,00 €

Stock real: 500,00

Este es el archivo generado:

```
catálogoHardware.csv
1 Producto,Precio de venta,Coste
2 Booking Fees,50.0,0.0
3 DUA VAT Valuation 10%,1.0,0.0
4 DUA VAT Valuation 21%,1.0,0.0
5 DUA VAT Valuation 4%,1.0,0.0
6 Figura Orco,50.0,0.0
7 Figura Pikachu,25.0,0.0
8 Procesador Intel Core i7-13700K,420.0,350.0
9 tornillos,1.0,0.0
10 |
```

Hito 2

Primero activo el modo Desarrollador
Nos metemos a Acciones de Servidor

Ajustes

Opciones generales

Usuarios y compañías

+

sgempresarial

4

Nuevo

Acciones de servidor

Acciones de nivel superior

Buscar...

<input type="checkbox"/> Nombre	Modelo	Tipo
<input type="checkbox"/> Enviar un correo	Pedido de venta	Ejecutar co
<input type="checkbox"/>	Proveedor de pago	Ejecutar co
<input type="checkbox"/> AI Agent Sources: Process Sources	Origen del agente de IA	Ejecutar co
<input type="checkbox"/> AI: Adjust Search	Agente IA	Ejecutar co
<input type="checkbox"/> AI: Compute Date	Agente IA	Ejecutar co
<input type="checkbox"/> AI: Compute Report Measures	Agente IA	Ejecutar co
<input type="checkbox"/> AI: Get Fields	Agente IA	Ejecutar co
<input type="checkbox"/> AI: Get Menu Details	Agente IA	Ejecutar co
<input type="checkbox"/> AI: Open Menu Graph	Agente IA	Ejecutar co
<input type="checkbox"/> AI: Open Menu Kanban	Agente IA	Ejecutar co
<input type="checkbox"/> AI: Open Menu List	Agente IA	Ejecutar co
<input type="checkbox"/> AI: Open Menu Pivot	Agente IA	Ejecutar co
<input type="checkbox"/> AI: Read group	Agente IA	Ejecutar co

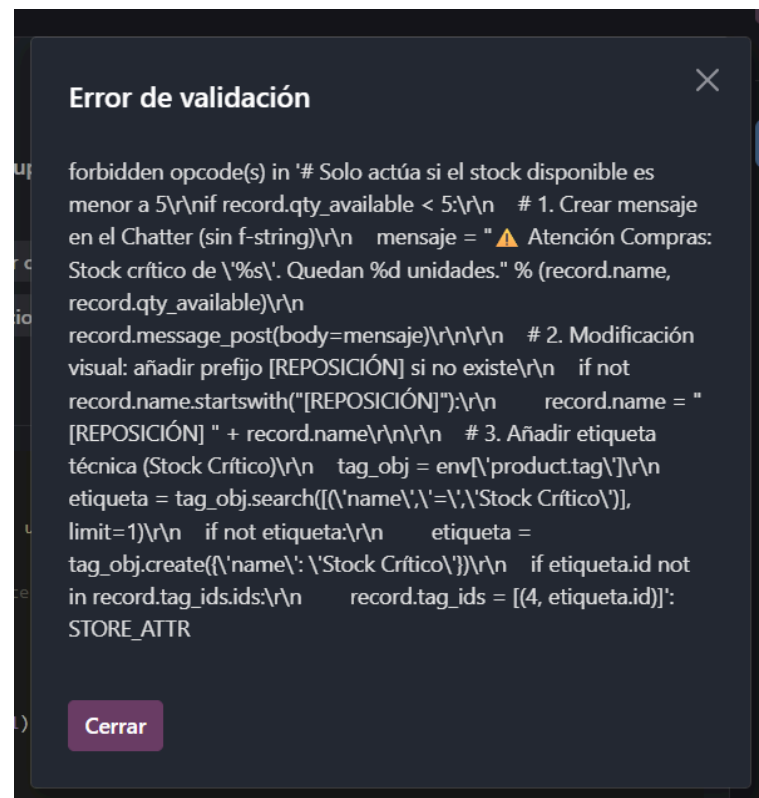
Código empleado:

```
# Solo actúa si el stock disponible es menor a 5
if record.qty_available < 5:
    # 1. Crear mensaje en el Chatter (sin f-string)
    mensaje = "⚠️ Atención Compras: Stock crítico de '%s'. Quedan %d unidades." % (record.name, record.qty_available)
    record.message_post(body=mensaje)

    # 2. Modificación visual: añadir prefijo [REPOSICIÓN] si no existe
    if not record.name.startswith("[REPOSICIÓN]"):
        record.name = "[REPOSICIÓN] " + record.name

    # 3. Añadir etiqueta técnica (Stock Crítico)
    tag_obj = env['product.tag']
    etiqueta = tag_obj.search([('name','=', 'Stock Crítico')], limit=1)
    if not etiqueta:
        etiqueta = tag_obj.create({'name': 'Stock Crítico'})
    if etiqueta.id not in record.tag_ids.ids:
        record.tag_ids = [(4, etiqueta.id)]
```

Me vuelve a dar error como en la actividad anterior



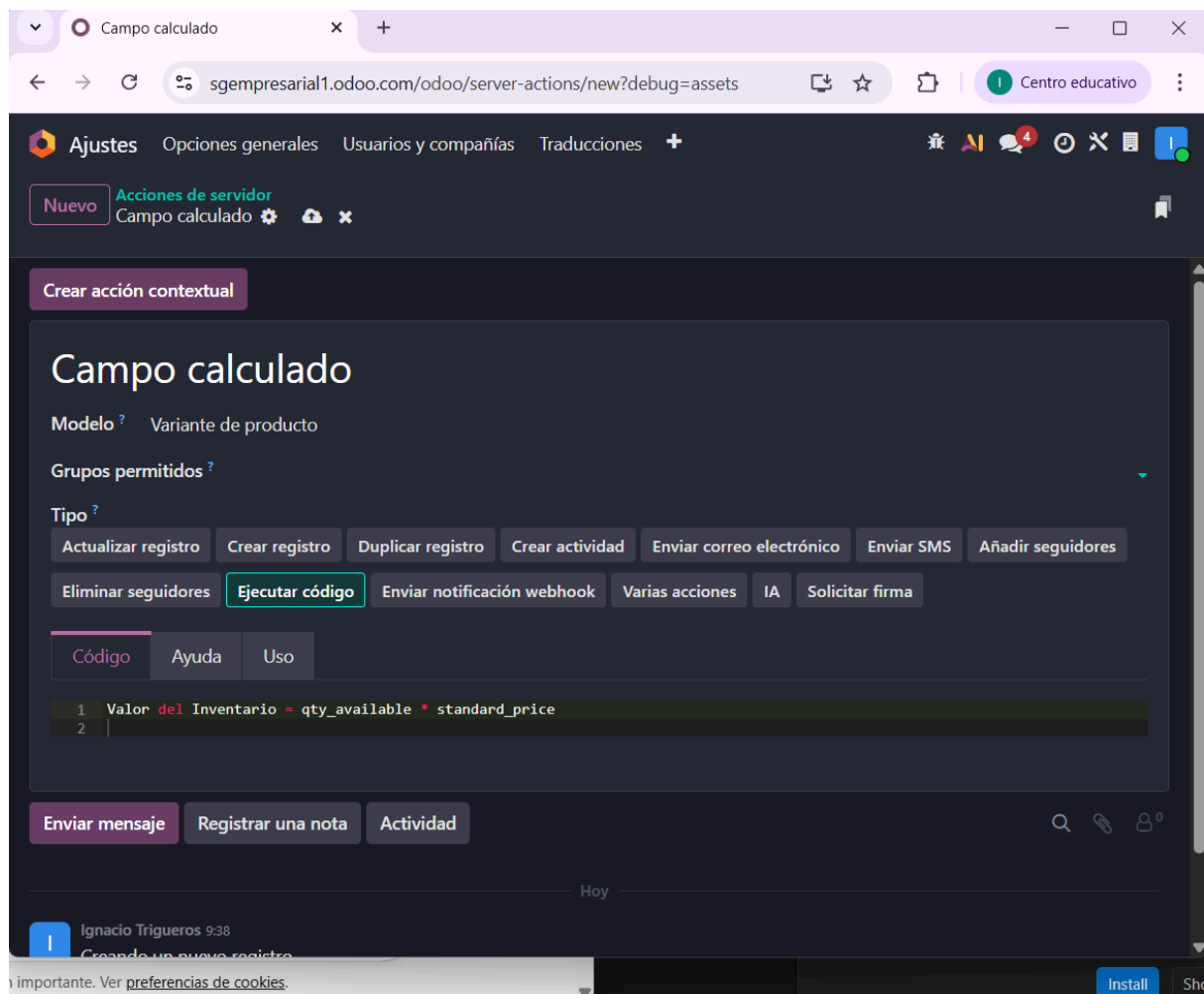
Hito 3

Cómo a partir de aquí no me deja, va a ser más un ejercicio teórico pero me da el mismo error para todos.

Creó un campo nuevo llamado Campo calculado, con el modelo product.product

Uso el script: **Valor del Inventario = qty_available * standard_price**

Preparo el entorno



Después de ver que no podía avanzar, lo he hecho más teórico, pero es que no me ha dado tiempo a más.

HITO 3: Business Intelligence (BI) y Dashboard de Valoración

El objetivo del Hito 3 es proporcionar a la dirección de Hardware Solutions S.L. una visión clara del capital inmovilizado en inventario, permitiendo identificar qué categorías de productos y qué almacenes concentran mayor inversión económica.

Para construir el KPI de Valor del Inventario no se recomienda calcular manualmente la fórmula cantidad disponible por coste unitario, ya que la cantidad disponible depende del contexto (almacén, fecha o compañía) y el coste estándar puede no reflejar correctamente métodos de valoración como FIFO o AVCO.

La forma correcta y profesional de obtener este KPI en Odoo es utilizar el reporte nativo "Valoración de Inventario", ubicado en el módulo de Inventario dentro del apartado de Reportes. Este informe se basa en el modelo técnico de capas de valoración de stock y calcula automáticamente el valor real contable de las existencias actuales.

Para construir el motor de visualización se debe cambiar la vista del reporte a formato Pivot (tabla dinámica), que permite organizar y analizar la información de forma multidimensional.

En la configuración de la tabla dinámica, las dimensiones deben organizarse agrupando en primer nivel por Almacén o Ubicación y en segundo nivel por Categoría de Producto. Esto permite analizar el valor del inventario siguiendo una jerarquía clara y lógica para la toma de decisiones.

Como métrica obligatoria se debe seleccionar el campo "Valor", que representa el valor monetario total del inventario según el método contable configurado en la empresa.

Una vez construida la tabla dinámica, se debe transformar en una visualización ejecutiva mediante la vista Gráfico, seleccionando el tipo de gráfico de barras para facilitar la interpretación rápida por parte de la dirección.

En el gráfico, el eje horizontal debe representar las categorías de producto y la medida debe ser el valor total del inventario. De manera opcional, se puede utilizar el almacén como filtro o como segunda dimensión para comparar ubicaciones.

Este dashboard permite detectar qué familias de productos, como CPUs, GPUs o placas base, concentran mayor inversión económica, facilitando decisiones estratégicas sobre compras, reposición y reducción de sobrestock.

Además, el análisis por almacén permite identificar posibles desequilibrios logísticos, como exceso de capital inmovilizado en una ubicación concreta.

El informe se actualiza automáticamente conforme cambian las existencias o los costes, garantizando que la dirección siempre disponga de información financiera actualizada y fiable.

Finalmente, la vista puede guardarse y compartirse con el equipo directivo, convirtiéndose en una herramienta permanente de apoyo para la planificación de compras y la optimización del capital de trabajo.

Git : <https://github.com/IgnacioGTri/ProyectoGestionFinal.git>

main

All users All time

Commits on Feb 13, 2026

- commit septimo**
ignaciotrigueros1993-cmyk committed 7 minutes ago eb18774

Commits on Feb 12, 2026

- commit sexto**
ignaciotrigueros1993-cmyk committed yesterday ebb7899
- comit cuarto?**
ignaciotrigueros1993-cmyk committed yesterday 8eb2f31

Commits on Feb 11, 2026

- cuarto commit**
IgnacioGTri committed 2 days ago a4cc287
- tercer commit**
IgnacioGTri committed 2 days ago 9d1c3b1
- segundo commit**
IgnacioGTri committed 2 days ago b5c3d8a
- primer commit**
IgnacioGTri committed 2 days ago e2300e7
- Initial commit**
IgnacioGTri authored 2 days ago Verified 0ab40eb

Reflexión técnica: Explica brevemente qué ventaja tiene usar una Acción de Servidor frente a que un empleado revise el stock manualmente cada mañana.

La automatización acelera el trabajo. Al final es una ayuda para el empleado, se acelera los procesos y además se garantiza que el sistema va a verificar stock de forma automática.

Al no pasar por manos humanas también previene fallos en los datos.

