



Guía de Integración - Custom Villagers

Para: Servidor Wetlands VoxeLibre **Fecha:** 2025-12-10 **Versión del Mod:** 1.0.0



Pasos de Integración Completos

Paso 1: Verificar Estructura del Mod

El mod ya está creado en:

```
/home/gabriel/Documentos/luanti-voxelibre-  
server/server/mods/custom_villagers/
```

Estructura actual:

```
custom_villagers/  
├── mod.conf                #  Creado  
├── init.lua                #  Creado  
├── README.md              #  Creado  
├── INTEGRATION_GUIDE.md   #  Este archivo  
├── textures/  
│   ├── README_TEXTURES.md #  Creado  
│   └── generate_placeholders.sh #  Creado (ejecutable)  
└── locale/  
    └── template.txt       #  Creado
```

Paso 2: Generar Texturas Placeholder

Opción A: Usar script automático (Recomendado)

```
# Navegar al directorio de texturas  
cd /home/gabriel/Documentos/luanti-voxelibre-  
server/server/mods/custom_villagers/textures/  
  
# Ejecutar generador  
bash generate_placeholders.sh
```

Resultado esperado:

- 20 archivos PNG (4 tipos × 5 caras)
- Colores distintivos: Verde (farmer), Azul (librarian), Morado (teacher), Marrón (explorer)

Opción B: Crear manualmente con ImageMagick

Ver instrucciones detalladas en [textures/README_TEXTURES.md](#)

Paso 3: Commit y Push al Repositorio

```
# Navegar al directorio raíz del proyecto
cd /home/gabriel/Documentos/luanti-voxelibre-server/

# Verificar archivos a agregar
git status

# Agregar mod completo
git add server/mods/custom_villagers/

# Commit descriptivo
git commit -m "🏠 Add: Custom Villagers mod v1.0.0 - Interactive NPCs
```

🎯 Características:

- 4 tipos de aldeanos (Farmer, Librarian, Teacher, Explorer)
- Sistema de diálogos educativos interactivos
- Sistema de comercio con esmeraldas
- Rutinas día/noche (activos de día, duermen de noche)
- Pathfinding básico cerca del hogar

🔧 Implementación:

- Compatible con VoxeLibre v0.90.1+
- Usa minetest.register_entity() para control total
- Formspeccs para interacciones
- Educativo y apropiado para niños 7+

📁 Archivos:

- init.lua - Sistema completo
- mod.conf - Configuración
- README.md - Documentación
- textures/ - Texturas placeholder
- locale/ - Template de traducciones

🧪 Testing:

- Probado localmente con sintaxis Lua correcta
- Compatible con estructura Docker del servidor

🤖 Generated with Claude Code

Co-Authored-By: Claude Sonnet 4.5 <noreply@anthropic.com>

```
# Push al repositorio remoto
git push origin main
```

Paso 4: Deployment en VPS

4.1 Conectar al VPS

```
ssh gabriel@167.172.251.27
```

4.2 Pull cambios del repositorio

```
# Navegar al directorio del proyecto
cd /home/gabriel/luanti-voxelibre-server

# Backup preventivo (recomendado)
docker-compose exec -T luanti-server cp -r /config/.minetest/worlds/world
/config/.minetest/worlds/world_backup_$(date +%Y%m%d_%H%M%S)

# Pull desde GitHub
git pull origin main
```

4.3 Verificar que el mod se descargó

```
ls -la server/mods/custom_villagers/

# Verificar archivos clave
ls -la server/mods/custom_villagers/init.lua
ls -la server/mods/custom_villagers/textures/*.png
```

Paso 5: Habilitar el Mod en VoxeLibre

```
# Opción 1: Agregar directamente al world.mt
docker-compose exec -T luanti-server sh -c 'echo "load_mod_custom_villagers
= true" >> /config/.minetest/worlds/world/world.mt'

# Opción 2: Editar manualmente
docker-compose exec luanti-server vi
/config/.minetest/worlds/world/world.mt
# Agregar línea: load_mod_custom_villagers = true

# Verificar que se agregó correctamente
docker-compose exec -T luanti-server cat
/config/.minetest/worlds/world/world.mt | grep custom_villagers
```

Salida esperada:

```
load_mod_custom_villagers = true
```

Paso 6: Reiniciar el Servidor

```
# Reiniciar contenedor de Luanti
docker-compose restart luanti-server

# Esperar inicio completo (30-40 segundos)
sleep 40

# Verificar que el servidor está corriendo
docker-compose ps | grep luanti-server
```

Estado esperado: Up

Paso 7: Verificar Carga del Mod

```
# Ver logs de inicialización
docker-compose logs --tail=50 luanti-server | grep -i "custom_villagers"
```

Mensajes esperados:

```
[custom_villagers] Initializing Custom Villagers v1.0.0
[custom_villagers] Registered villager type: farmer
[custom_villagers] Registered villager type: librarian
[custom_villagers] Registered villager type: teacher
[custom_villagers] Registered villager type: explorer
[custom_villagers] Custom Villagers v1.0.0 loaded successfully!
```

Paso 8: Testing en el Juego

8.1 Conectar al servidor

Desde el cliente Luanti:

- **Dirección:** `luanti.gabrielpantoja.cl`
- **Puerto:** `30000`

8.2 Obtener privilegios de server (si es necesario)

```
/grant <tu_nombre> server
```

8.3 Spawnear aldeanos de prueba

```
/spawn_villager farmer  
/spawn_villager librarian  
/spawn_villager teacher  
/spawn_villager explorer
```

8.4 Verificar funcionalidad

- ☐ **Aldeanos aparecen visualmente** (con texturas de colores)
- ☐ **Click derecho abre menú de interacción**
- ☐ **Diálogos funcionan** (botones Saludar, Trabajo, Educación)
- ☐ **Comercio funciona** (si está habilitado)
- ☐ **Aldeanos caminan** cerca de su hogar
- ☐ **Rutinas día/noche** (duermen de noche)

Configuración Opcional

Ajustar configuración en server/config/luanti.conf

```
# Agregar al final del archivo luanti.conf  
  
# === Custom Villagers Configuration ===  
  
# Máximo de aldeanos por área  
custom_villagers_max_villagers = 5  
  
# Radio de spawn natural  
custom_villagers_spawn_radius = 20  
  
# Habilitar debug (solo para desarrollo)  
custom_villagers_debug = false  
  
# Habilitar comercio  
custom_villagers_enable_trading = true  
  
# Habilitar rutinas de horarios  
custom_villagers_enable_schedules = true
```

Aplicar cambios:

```
docker-compose restart luanti-server
```

Mejoras Post-Integración

Fase 1: Texturas Mejoradas

1. Crear texturas pixel art personalizadas

- Usar GIMP, Krita o Aseprite
- Resolución 16x16 o 32x32
- Seguir guía en [textures/README_TEXTURES.md](#)

2. Reemplazar placeholders

```
# Local: editar texturas  
# Git add, commit, push  
# VPS: git pull y restart
```

Fase 2: Modelos 3D (Avanzado)

1. Crear modelos .b3d con Blender

2. Modificar init.lua para usar mesh:

```
visual = "mesh",  
mesh = "custom_villagers_farmer.b3d",
```

Fase 3: Sonidos

1. Agregar efectos de voz/ambiente (.ogg)

2. Implementar en on_rightclick

Troubleshooting Común

Problema 1: Mod no se carga

Síntomas: No aparece en logs, comandos no funcionan

Diagnóstico:

```
# Verificar que el mod está en el directorio correcto  
docker-compose exec -T luanti-server ls -la /config/.minetest/mods/ | grep  
custom_villagers
```

```
# Verificar habilitación en world.mt
docker-compose exec -T luanti-server cat
/config/.minetest/worlds/world/world.mt | grep custom_villagers

# Revisar errores de sintaxis Lua
docker-compose logs luanti-server | grep -i error | grep -i
custom_villagers
```

Soluciones:

1. Verificar `load_mod_custom_villagers = true` en `world.mt`
2. Reiniciar servidor
3. Verificar sintaxis Lua con `lua -c init.lua`

Problema 2: Texturas faltantes (cubos negros/blancos)

Síntomas: Aldeanos aparecen pero sin colores

Diagnóstico:

```
# Verificar texturas en contenedor
docker-compose exec -T luanti-server ls -la
/config/.minetest/mods/custom_villagers/textures/

# Verificar cantidad de archivos PNG
docker-compose exec -T luanti-server find
/config/.minetest/mods/custom_villagers/textures/ -name "*.png" | wc -l
```

Esperado: 20 archivos PNG

Soluciones:

1. Generar placeholders con `generate_placeholders.sh`
2. Verificar nombres exactos de archivos (case-sensitive)
3. Git pull para asegurar sincronización

Problema 3: Aldeanos no aparecen al usar comando

Síntomas: Comando ejecuta pero no hay entity

Diagnóstico:

```
# Verificar logs de spawn
docker-compose logs --tail=20 luanti-server
```

Soluciones:

1. Verificar que tienes privilegio `server`
 2. Usar nombres correctos: farmer, librarian, teacher, explorer (lowercase)
 3. Verificar que no hay errores de registro de entidad
-

Problema 4: Diálogos en blanco

Síntomas: Menú se abre pero no hay texto

Causa: Base de datos de diálogos no inicializada

Solución: Reiniciar servidor (los diálogos están hardcoded en init.lua)



Métricas de Éxito

Después de la integración, verifica:

- ☒ **Mod cargado:** Aparece en logs de inicialización
 - ☒ **4 tipos registrados:** farmer, librarian, teacher, explorer
 - ☒ **Comandos funcionan:** /spawn_villager, /villager_info
 - ☒ **Entidades visibles:** Aldeanos con texturas de colores
 - ☒ **Interacción OK:** Click derecho abre menú
 - ☒ **Diálogos OK:** Mensajes educativos aparecen
 - ☒ **Comercio OK:** Transacciones funcionan (si está habilitado)
 - ☒ **Comportamiento OK:** Caminan y duermen de noche
-



Próximos Pasos

Educación y Comunidad

1. Crear tutorial en el juego

- Libro explicativo sobre aldeanos
- Comando `/tutorial_villagers`

2. Integrar con otros mods

- Compatibilidad con `education_blocks`
- Sistema de misiones educativas

3. Comunidad de jugadores

- Anunciar nueva característica en Discord
 - Recopilar feedback de niños jugadores
 - Iterar mejoras basadas en uso real
-



Soporte y Contacto

Documentación completa: [README.md](#) **Texturas:** [textures/README_TEXTURES.md](#) **Modding general:**
[../.. / docs / mods / MODDING_GUIDE.md](#)

Repositorio: <https://github.com/gabrielpantoja-cl/luanti-voxelibre-server> **VPS:** 167.172.251.27 **Servidor de juego:** luanti.gabrielpantoja.cl:30000

¡Integración lista para deployment! 🚀🏠