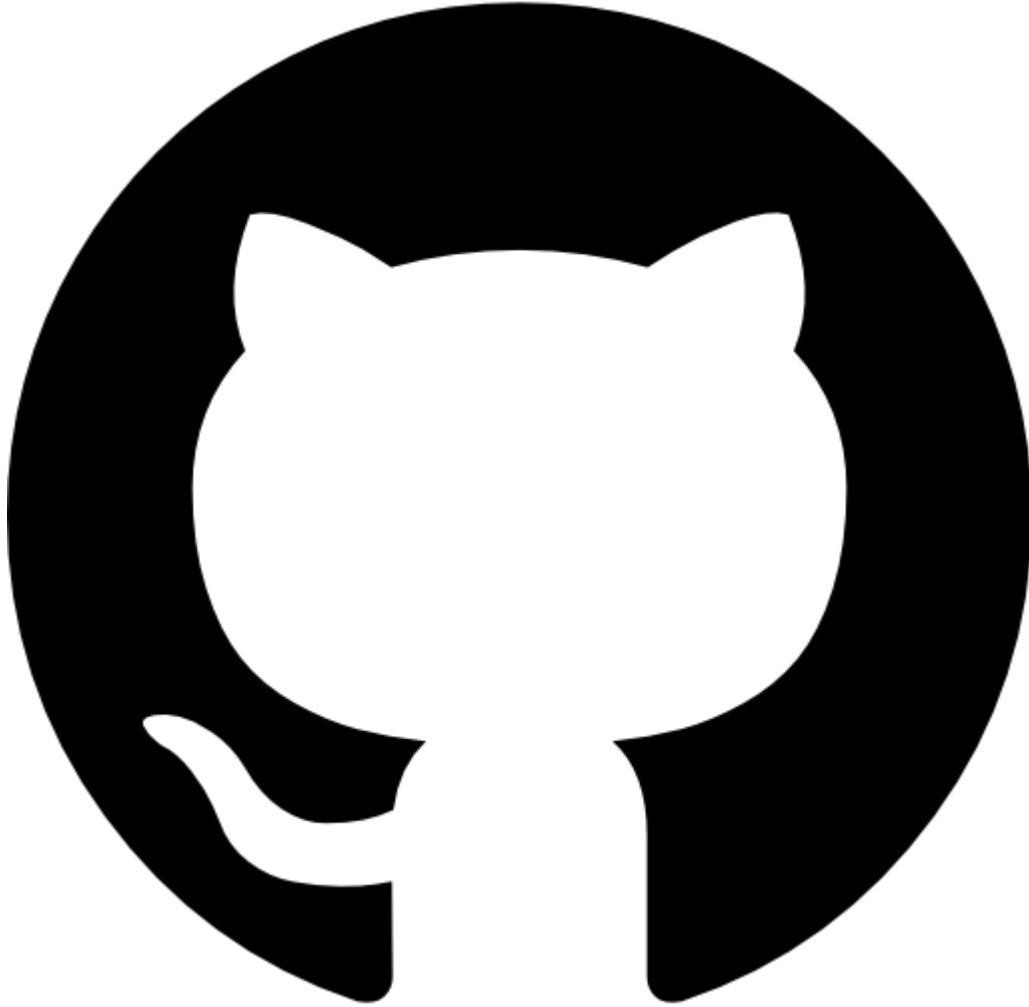


**Proyecto:** Repositorio colaborativo de Mods en Minecraft.

**Autores:** Berta Pena, Mario Fuertes, Armando Shuguli



[https://github.com/MarioFuertesLopez/Minecraft\\_mods-plugins/tree/main](https://github.com/MarioFuertesLopez/Minecraft_mods-plugins/tree/main)

**Fecha:** 27 de enero de 2026

**Asignatura:** Complementos de Desarrollo del Software

<b>1. Introducción .....</b>	<b>3</b>
<b>2. Descripción del proyecto .....</b>	<b>4</b>
<b>3. Tecnologías empleadas .....</b>	<b>6</b>
3.1 Control de versiones	
3.2 Plataforma de juego	
3.3 Frameworks de modding	
3.4 Documentación	
<b>4. Estructura del repositorio .....</b>	<b>7</b>
<b>5. Documentación del repositorio .....</b>	<b>9</b>
5.1 Archivo README.md principal	
5.2 README.md en subcarpetas	
5.3 Guías adicionales	
<b>6. Control de versiones .....</b>	<b>11</b>
<b>7. Pruebas y validación .....</b>	<b>13</b>
7.1 Verificación de mods	
7.2 Validación de documentación	
<b>8. Gestión del proyecto con Issues .....</b>	<b>14</b>
8.1 Uso de Issues	
8.2 Categorización de Issues	
8.3 Flujo de trabajo con Issues	
8.4 Documentación de bugs	
8.5 Beneficios del sistema de Issues	
<b>9. Colaboración comunitaria .....</b>	<b>16</b>
9.1 Modelo colaborativo	
9.2 Directrices para colaboradores	
9.3 Revisión de contribuciones	
<b>10. Conclusión .....</b>	<b>17</b>
<b>11. Anexos .....</b>	<b>18</b>
Capturas del repositorio	

## 1.Introducción

Este proyecto consiste en la creación y gestión de un repositorio colaborativo en GitHub cuyo objetivo es almacenar, organizar y documentar mods de Minecraft para facilitar la experiencia al jugador principiante.

Minecraft es uno de los videojuegos más populares del mundo, con una comunidad activa que desarrolla miles de modificaciones (mods) que amplían y mejoran la experiencia de juego. Sin embargo, para usuarios novatos, el proceso de encontrar, descargar e instalar mods puede resultar confuso y abrumador

- La gran cantidad de mods disponibles en múltiples plataformas
- Problemas de compatibilidad entre mods
- Diferentes versiones de Minecraft (Java y Bedrock)
- Múltiples actualizaciones del juego (1.19, 1.20, 1.21, etc.)
- Diferentes loaders de mods (Forge, Fabric, NeoForge)
- Falta de documentación clara y centralizada

Es muy habitual para los usuarios de Minecraft el uso de mods. Sin embargo, la cantidad de mods que existen para este juego es abrumadora. Además, no solo es un problema de que varios mods pueden hacer una función similar, sino que pueden entrar en conflicto entre ellos.

Este repositorio nace como solución a estos problemas, proporcionando una colección curada y organizada de mods esenciales con documentación clara para usuarios principiantes.

## 2.Descripción del proyecto

Este repositorio tiene como finalidad:

- **Centralizar** distintos mods organizados por versión y loader
- **Facilitar** la instalación y uso de los mods mediante guías claras
- **Aplicar** una estructura ordenada y mantenible al proyecto
- **Permitir** la colaboración de la comunidad de forma segura

Cada mod se organiza en su repositorio que incluye:

### Versiones soportadas:

- Minecraft Java Edition (versiones 1.0 - 1.21)
- Subdivisiones específicas para versiones con cambios significativos (ej: 1.20.0 - 1.20.6)

### Plataformas de modding incluidas:

- **Vanilla:** Para recursos básicos sin necesidad de loaders
- **Fabric:** Loader ligero enfocado en rendimiento y utilidades
- **NeoForge:** Evolución de Forge, ideal para mods de contenido masivo
- **Shaders:** Modificaciones visuales compatibles con OptiFine/Iris

El repositorio se enfoca en mods esenciales para usuarios principiantes, organizados en categorías:

1. **Optimización:** Mejoran el rendimiento del juego

- Sodium (Fabric) - Mejora drástica de FPS
  - Lithium (Fabric) - Optimización del servidor
  - OptiFine - Optimización y shaders para Forge/Vanilla
2. **Mapas y navegación:**
- Xaero's World Map - Mapa del mundo completo
  - Xaero's Minimap - Minimapa en pantalla
  - JourneyMap - Alternativa de mapa con waypoints
3. **Interfaz y utilidades:**
- JEI (Just Enough Items) - Recetas y búsqueda de objetos
  - AppleSkin - Información de comida y saturación
  - Inventory Tweaks - Mejoras en el inventario
4. **Contenido:**
- Biomes O'Plenty - Nuevos biomas
  - Alex's Mobs - Nuevas criaturas
  - Chisels & Bits - Construcción detallada a nivel de píxel

También hay que tener en cuenta:

- La **versión de Minecraft** (Java o Bedrock)
- La **actualización específica** (1.20.1, 1.19.2, etc.)
- El **loader utilizado** (Forge, Fabric, NeoForge)
- Las **dependencias entre mods** (algunos requieren otros para funcionar)

Obviamente hay muchas cosas a tener en cuenta: qué mods utilizar, dónde conseguirlos, qué versión descargar y cómo instalarla. El objetivo de este repositorio es dar una solución organizada a este problema para que aquellos que entran sin muchos conocimientos en este mundo puedan jugar usando mods sin demasiados problemas.

### 3. Tecnologías empleadas

#### 3.1 Control de versiones

- **Git:** Sistema de control de versiones distribuido que permite rastrear cambios, trabajar en ramas y colaborar eficientemente
- **GitHub:** Plataforma de alojamiento del repositorio que facilita la colaboración mediante Issues, Pull Requests y documentación

#### 3.2 Plataforma de juego

- **Minecraft Java Edition:** Versión del juego compatible con mods (a diferencia de Bedrock Edition que tiene limitaciones)

#### 3.3 Frameworks de modding

- **Forge/NeoForge:** Framework tradicional para mods de contenido masivo
- **Fabric:** Framework moderno y ligero enfocado en rendimiento

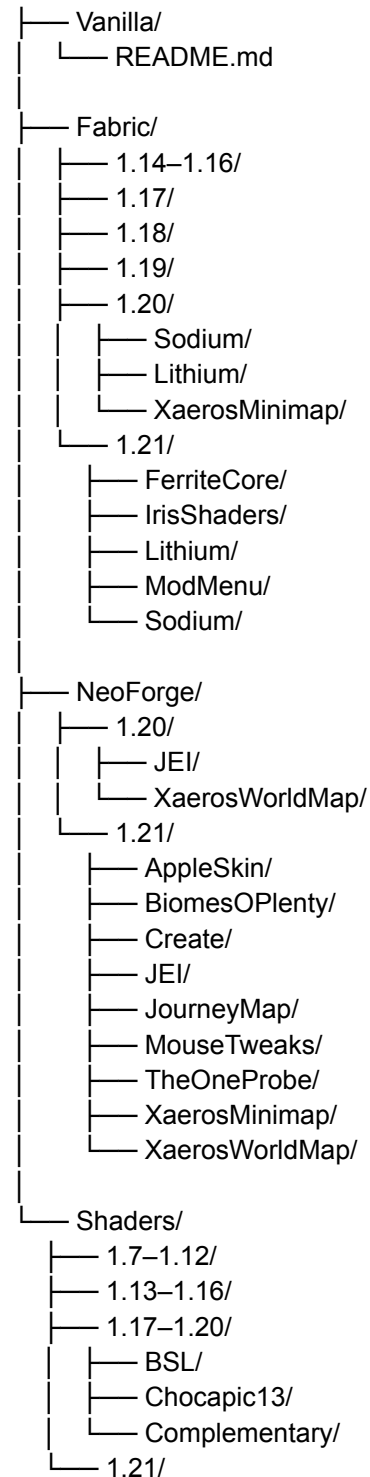
#### 3.4 Documentación

- **Markdown (.md):** Formato utilizado para README, guías y documentación del proyecto

#### 4. Estructura del repositorio

El repositorio contiene una colección de mods desarrollados por terceros con su respectiva documentación sobre su uso y organizados jerárquicamente para facilitar la navegación y evitar incompatibilidades.

Java/



Organización por loader primero:

El usuario generalmente sabe qué loader utiliza (o puede identificarlo fácilmente)

Evita confusión al navegar entre diferentes frameworks

Organización por mod:

Cada mod tiene su propia carpeta con:

- Archivo .jar del mod
- README.md con descripción, requisitos y enlace oficial

Agrupación de versiones antiguas:

Las versiones muy antiguas (1.0-1.5, 1.6-1.8) se agrupan ya que tienen menos usuarios activos

Estas carpetas están disponibles bajo demanda mediante Issues

Sección Vanilla:

Actualmente en construcción. Vanilla no soporta mods .jar pero sí datapacks y resource packs que se añadirán próximamente.

## **5. Documentación del repositorio**

### **Archivo README.md principal**

El README del repositorio es el documento más importante y la primera impresión para cualquier usuario. Incluye:

#### **Descripción general:**

- Qué es el repositorio y su propósito
- Público objetivo (usuarios novatos)
- Beneficios de usar este repositorio

#### **Guía de loaders:**

- Explicación clara de qué son Vanilla, Fabric y NeoForge
- Comparativa de ventajas y desventajas
- Ayuda para elegir el loader apropiado según necesidades

#### **Instrucciones de uso:**

- Cómo identificar la versión de Minecraft instalada
- Cómo determinar qué loader se está usando
- Navegación por la estructura del repositorio

#### **Proceso de instalación:**

- Descarga de archivos .jar
- Localización de la carpeta .minecraft según sistema operativo
- Copiado de archivos a la carpeta mods
- Inicio del juego con los mods cargados

#### **Reglas de contribución:**

- Organización requerida para subir mods

- Archivos permitidos y prohibidos
- Información obligatoria a incluir
- Respeto a licencias y autoría

### **README.md en subcarpetas**

Cada carpeta de versión específica incluye su propio README con:

- Lista de mods disponibles en esa carpeta
- Versión específica de cada mod
- Breve descripción de funcionalidad
- Autor original y enlace a fuente oficial
- Dependencias si las hay

### **5.3 Guías adicionales**

El repositorio incluye (o incluirá) documentación adicional:

- **INSTALLATION.md:** Guía detallada con capturas de pantalla
- **CONTRIBUTING.md:** Reglas completas para colaboradores
- **FAQ.md:** Preguntas frecuentes y solución de problemas

## **6. Control de versiones**

Los commits siguen una filosofía **atómica y descriptiva**:

### **Commits atómicos:**

- Cada commit representa un cambio lógico único
- Facilita la revisión de historial
- Permite revertir cambios específicos sin afectar otros

### **Mensajes descriptivos:**

- Formato: `[Categoría] Descripción breve del cambio`
- Ejemplos:
  - `[Docs] Actualizar README con sección de loaders`
  - `[Mods] Añadir Sodium 0.5.8 para Fabric 1.20`
  - `[Fix] Corregir enlace roto en CONTRIBUTING.md`

### **Uso de ramas (branches)**

El repositorio utiliza una estrategia de ramificación para mantener estabilidad:

### **Rama principal (main):**

- Contiene solo código estable y probado
- Protegida contra commits directos
- Solo se actualiza mediante Pull Requests aprobados

### **Ramas de funcionalidad (feature/\*):**

- `feature/add-fabric-mods` - Añadir nuevos mods de Fabric
- `feature/update-documentation` - Mejorar documentación
- `feature/reorganize-structure` - Cambios en estructura

### **Ramas de corrección (fix/\*):**

- `fix/broken-links` - Corregir enlaces rotos
- `fix/mod-compatibility` - Solucionar incompatibilidades

### **Proceso de fusión (merge)**

#### **Pull Requests:**

1. Creación de rama desde main
2. Desarrollo de cambios en la rama
3. Apertura de Pull Request con descripción clara
4. Revisión por colaboradores
5. Resolución de conflictos si existen
6. Aprobación y merge a main

#### **Resolución de conflictos:**

- Identificación de archivos en conflicto
- Revisión manual de cambios
- Decisión sobre qué versión mantener
- Prueba de funcionamiento post-merge

## **7. Pruebas y validación**

### **7.1 Verificación de mods**

Antes de incluir un mod en el repositorio, se realiza:

#### **Pruebas de funcionalidad:**

- Instalación del mod en entorno de prueba
- Verificación de que carga correctamente
- Comprobación de funcionalidad básica

#### **Pruebas de compatibilidad:**

- Verificar versión exacta de Minecraft soportada
- Comprobar compatibilidad con el loader especificado
- Probar junto a otros mods comunes para detectar conflictos

#### **Verificación de seguridad:**

- Descarga solo de fuentes oficiales (CurseForge, Modrinth, etc.)
- Revisión de archivos sospechosos



- Exclusión de mods con malware conocido

## 7.2 Validación de documentación

La documentación se valida mediante:

- Prueba de todos los enlaces incluidos
- Verificación de que las instrucciones son claras y completas
- Revisión ortográfica y gramatical
- Comprobación de formato Markdown correcto

## 8. Gestión del proyecto con Issues

### 8.1 Uso de Issues

Los **Issues** son una herramienta fundamental de GitHub que permite registrar:

- Tareas pendientes
- Errores detectados (bugs)
- Propuestas de mejora
- Solicitudes de nuevos mods

### 8.2 Categorización de Issues

**Labels (etiquetas):**

- **bug** - Errores en el repositorio
- **enhancement** - Mejoras propuestas
- **documentation** - Relacionado con documentación
- **new-mod** - Solicitud de añadir nuevo mod
- **help-wanted** - Se necesita ayuda de la comunidad

### 8.3 Flujo de trabajo con Issues

1. **Creación:** Usuario o colaborador crea Issue describiendo el problema/solicitud
2. **Asignación:** Se asigna a un colaborador responsable
3. **Desarrollo:** El colaborador trabaja en una rama específica
4. **Referencia:** Los commits mencionan el Issue (ej: **Fix #15: Corregir enlace roto**)
5. **Cierre:** El Issue se cierra automáticamente al hacer merge del Pull Request

### 8.4 Documentación de bugs

Los bugs se documentan con:

- Descripción clara del problema
- Pasos para reproducir el error
- Comportamiento esperado vs. comportamiento actual
- Versión afectada
- Capturas de pantalla si aplica

## 8.5 Beneficios del sistema de Issues

- **Centralización:** Toda la información en un solo lugar
- **Trazabilidad:** Historial completo de problemas y soluciones
- **Comunicación:** Facilita la colaboración entre miembros
- **Organización:** Priorización y seguimiento de tareas

## 9. Colaboración comunitaria

### 9.1 Modelo colaborativo

El repositorio está diseñado como proyecto de código abierto:

- Cualquiera puede contribuir mediante Pull Requests
- Las contribuciones son revisadas antes de aceptarse
- Se fomenta la participación de la comunidad

### 9.2 Directrices para colaboradores

#### Requisitos técnicos:

- Conocimientos básicos de Git y GitHub
- Comprensión de la estructura del repositorio
- Respeto a las normas de organización

#### Requisitos éticos:

- Respetar licencias de los mods
- Dar crédito a autores originales
- Solo incluir contenido legítimo y seguro

### 9.3 Revisión de contribuciones

Todas las contribuciones pasan por:

1. Revisión técnica (estructura correcta, archivos apropiados)
2. Revisión de seguridad (verificar fuente del mod)
3. Revisión de documentación (README actualizado)
4. Aprobación final antes de merge

## 10. Conclusión

Este proyecto ha permitido aplicar de forma práctica los conceptos de control de versiones, trabajo colaborativo y gestión de proyectos software en un caso de uso real y útil para la comunidad de jugadores de Minecraft.

Se ha logrado crear un repositorio bien estructurado, con documentación completa y un sistema de colaboración funcional que facilita la contribución comunitaria. La organización jerárquica por loader y versión resuelve efectivamente el problema de compatibilidad que enfrentan los usuarios novatos.

El uso de Git y GitHub no solo ha permitido mantener un historial claro de cambios, sino que también ha facilitado la implementación de un flujo de trabajo profesional mediante branches, Pull Requests e Issues.

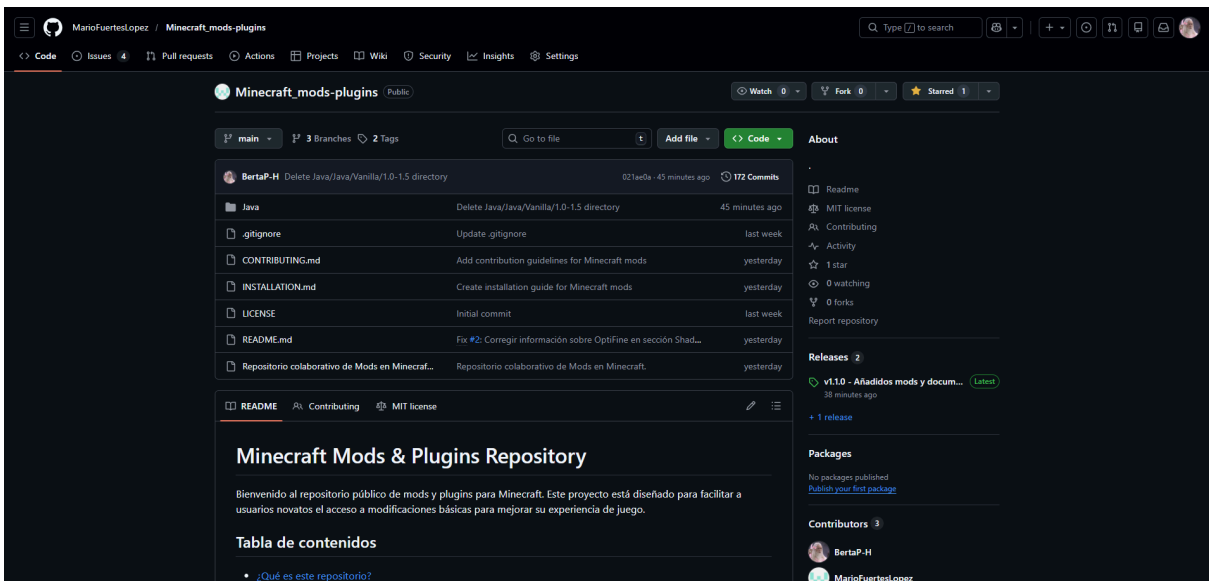
Más allá de los aspectos técnicos, el proyecto ha demostrado la importancia de la documentación clara y la consideración del usuario final al diseñar sistemas. La decisión de priorizar la simplicidad para usuarios novatos sobre la complejidad técnica ha resultado en un producto más accesible y útil.

Este repositorio representa un ejemplo práctico de cómo las herramientas de desarrollo de software pueden aplicarse más allá del código, creando soluciones organizativas que benefician a la comunidad.

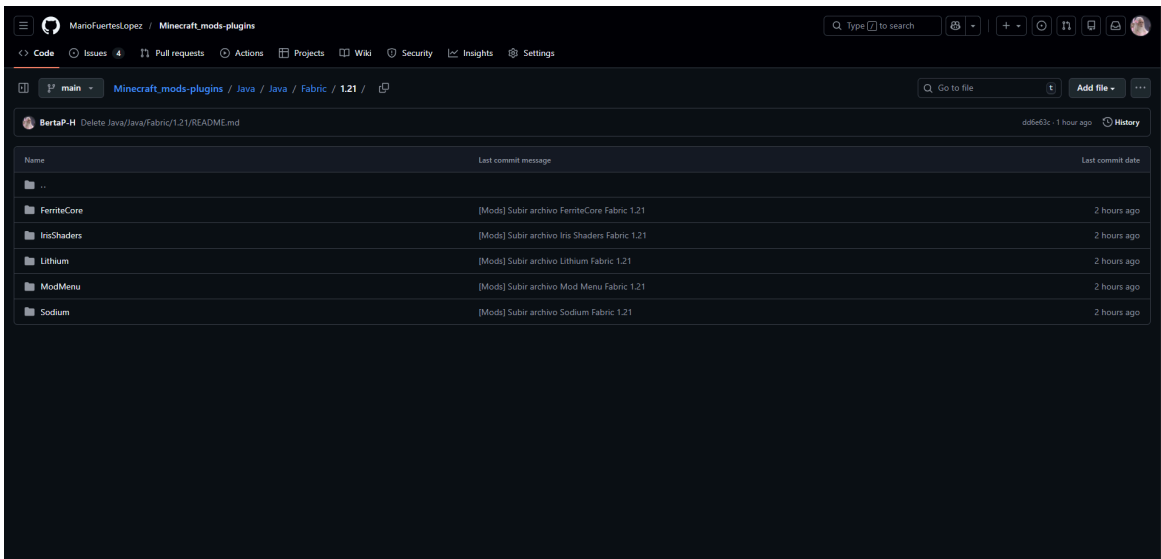
## 11. Anexos

### Capturas del repositorio

#### 11.1 - Página principal del repositorio



#### 11.2 - Estructura de carpetas (Fabric 1.21)



#### 11.3 - Historial de commits

Commits

main

All usersAll time

Commits on Jan 27, 2026

Delete Java/Java/Vanilla/1.0-1.5 directory	Verified	021ae0a	<>
Delete Java/Java/Vanilla/1.13-1.16 directory	Verified	b9d7ec4	<>
Delete Java/Java/Vanilla/1.17 directory	Verified	93a52c6	<>
Delete Java/Java/Vanilla/1.18 directory	Verified	36158c8	<>
Delete Java/Java/Vanilla/1.19 directory	Verified	f0fa431	<>
Delete Java/Java/Vanilla/1.21/README.md	Verified	3ee0cb	<>
Delete Java/Java/Vanilla/1.21/snapshots directory	Verified	d40d7be	<>
Delete Java/Java/Vanilla/1.21/releases/README.md	Verified	f353950	<>
Delete Java/Java/Vanilla/1.20/1.20.1/README.md	Verified	01298be	<>
Delete Java/Java/Vanilla/1.20/1.20.0/README.md	Verified	6e4837a	<>
Delete Java/Java/Vanilla/1.20/1.20.2/README.md	Verified	9b4e42a	<>

11.4 - Pull Requests cerrados

MarioFuentesLopez / Minecraft\_mods-plugins

Type to search

CodeIssues4Pull requestsActionsProjectsWikiSecurityInsightsSettings

Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

Dismiss

Filtersispr is:closed

Labels0Milestones0New pull request

Clear current search query, filters, and sorts

0 Open3 Closed

AuthorLabelProjectsMilestonesReviewsAssigneeSort

Fix #2: Corregir información sobre OptiFine en sección Shaders

#7 by BertalP-H was merged yesterday

Add FAQ section to README

#2 by BertalP-H was merged yesterday

Create installation guide for Minecraft mods

#1 by BertalP-H was merged yesterday

2

ProTip! Type g on any issue or pull request to go back to the pull request listing page.

© 2026 GitHub, Inc. TermsPrivacySecurityStatusCommunityDocsContactManage cookiesDo not share my personal information

11.5 - Issues del proyecto

MarioFuentesLopez / Minecraft\_mods-plugins

Type to search

CodeIssues4Pull requestsActionsProjectsWikiSecurityInsightsSettings

Issue state:open

LabelsMilestonesNew issue

Open4Closed1

AuthorLabelsProjectsMilestonesAssigneesF Newest

Crear README específicos para cada carpeta de versión

documentationhelp wanted

#6 BertalP-H opened yesterday

Añadir capturas de pantalla a la guía de instalación

documentationenhancement

#5 BertalP-H opened yesterday

Enlace roto en sección de Shaders del README

bugdocumentation

#4 BertalP-H opened yesterday

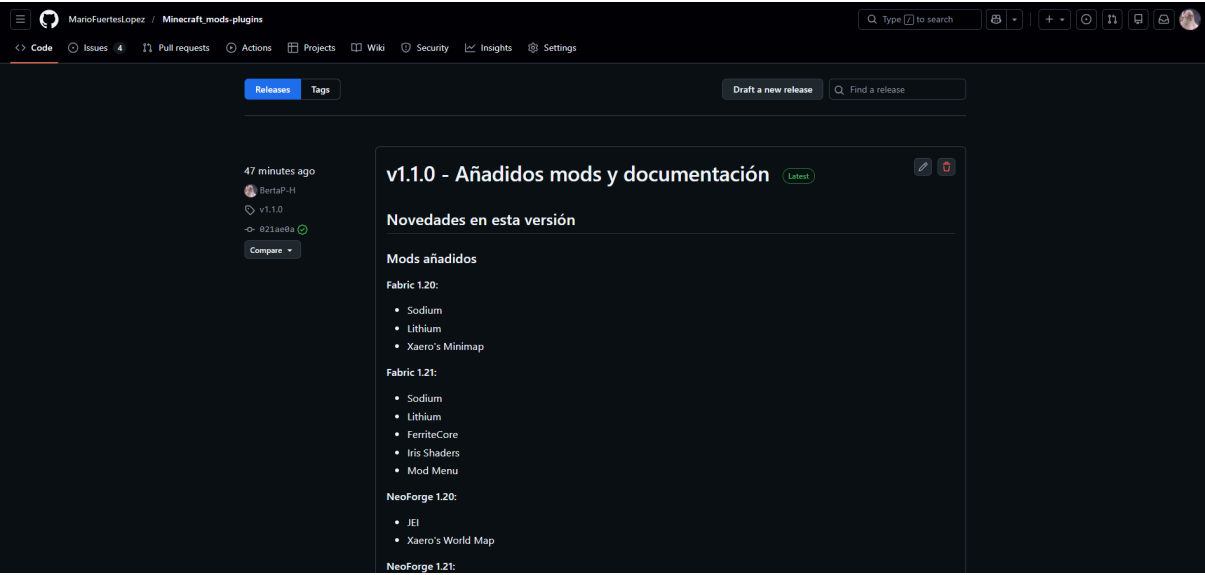
Añadir Xaero's World Map

enhancement

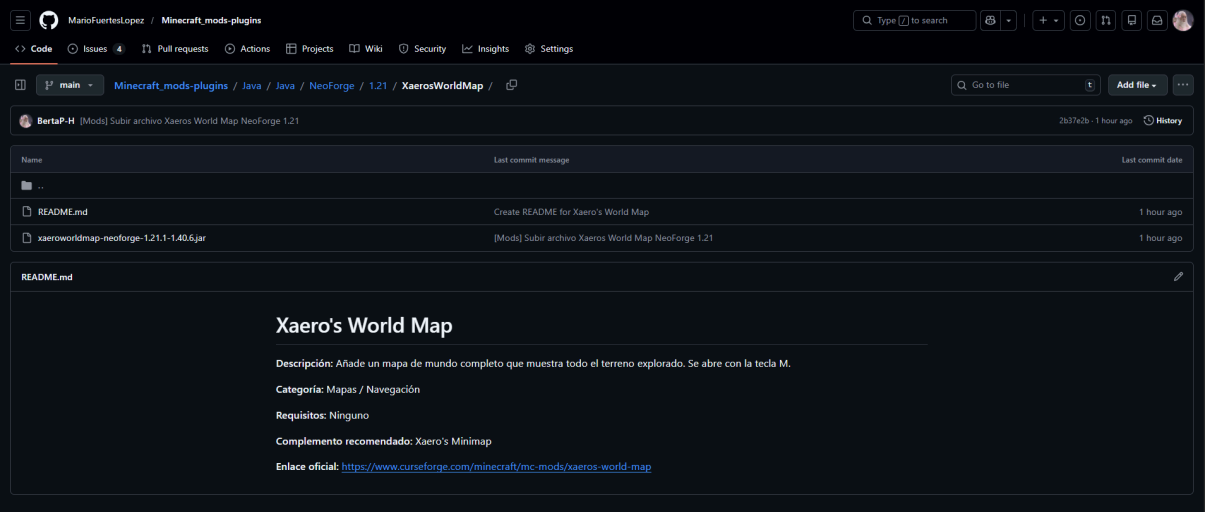
#3 BertalP-H opened yesterday

Give feedback

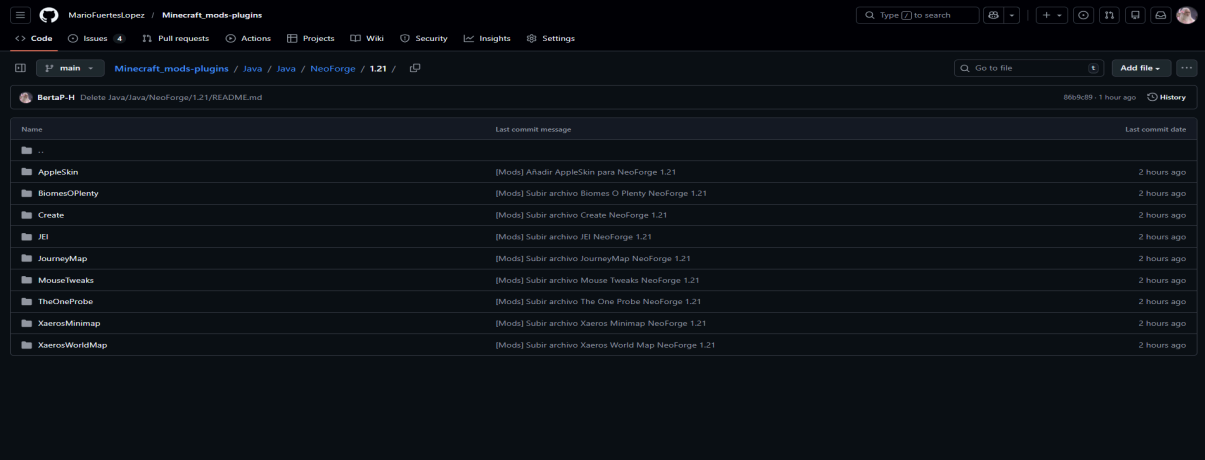
11.6 - Releases publicadas



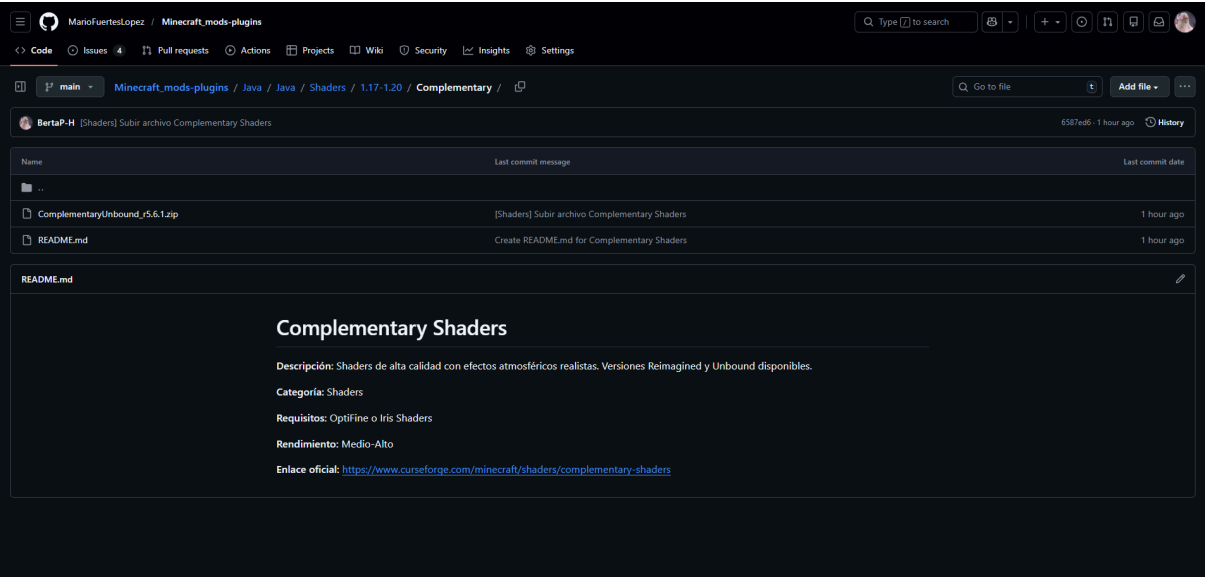
## 11.7 - Ejemplo de mod con README (Xaero's World Map)



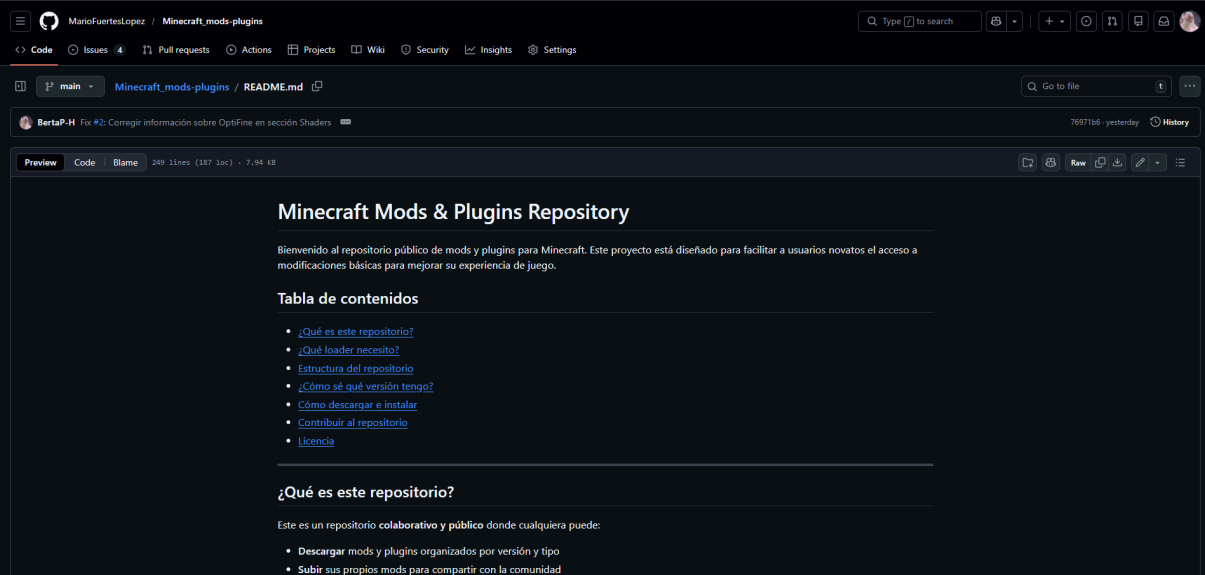
## 11.8 - Estructura de carpetas (NeoForge 1.21)



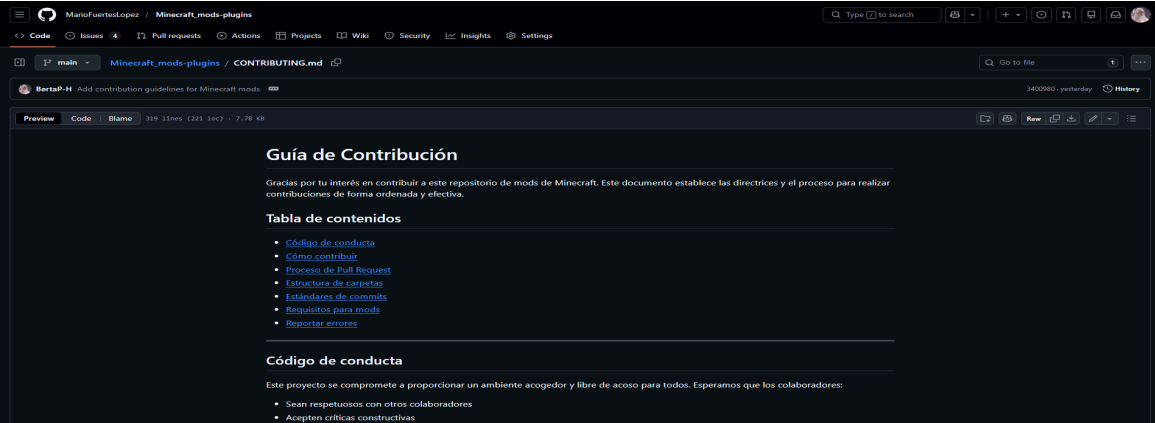
## 11.9 - Ejemplo de shader con README (BSL)



## 11.10 - README principal del repositorio



## 11.11 - Archivo CONTRIBUTING.md



## 11.12 - Archivo INSTALLATION.md

