



Práctica 3. Creación de mod y mappings

Introducción

En la versión original de Minecraft Java Edition, los nombres de los parámetros son muy abstractos y difíciles de interpretar. Para solucionarlo, vamos a usar *parchment mappings* para poder cambiar los nombres de los parámetros.

Adicionalmente, crearemos **nuestro primer mod**, el cual nos permitirá tener una base para hacer futuras modificaciones del juego, por lo que no tendrá ninguna funcionalidad real en el juego.

Los Mods de Minecraft son un complemento que se añade al juego para efectuar modificaciones (mod de “modification”) para mejorar su jugabilidad/funcionalidad.

Mappings

Modificad el fichero *build.gradle* que se encuentra dentro de nuestro proyecto, según los pasos que se describen a continuación. Información extraída de <https://github.com/ParchmentMC/Librarian/blob/dev/docs/FORGEGRADLE.md>.

```
21 # see the associated build.gradle script for how to populate this comple
22 version="${file.jarVersion}" #mandatory
23 # A display name for the mod
24 displayName="Example Mod" #mandatory
25 # A URL to query for updates for this mod. See the JSON update specification https
26 #updateJSONURL="https://change.me.example.invalid/updates.json" #optional
27 # A URL for the "homepage" for this mod, displayed in the mod UI
28 #displayURL="https://change.me.to.your.mods.homepage.example.invalid/" #optional
29 # A file name (in the root of the mod JAR) containing a logo for display
30 logoFile="examplemod.png" #optional
31 # A text field displayed in the mod UI
32 credits="Thanks for this example mod goes to Java" #optional
33 # A text field displayed in the mod UI
34 authors="Love, Cheese and small house plants" #optional
35 # The description text for the mod (multi line!) (#mandatory)
36 description=''
37 This is a long form description of the mod. You can write whatever you want here
38
39 Have some lorem ipsum.
40
41 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed mollis lacinia magna.
42 ...
43 # A dependency - use the . to indicate dependency for a specific modid. Dependenci
44 [[dependencies.examplemod]] #optional
45 # the modid of the dependency
46 modId="forge" #mandatory
```



Proyecto ICAI - MINECRAFT



Pasos a seguir:

1. Añadir “maven { url = 'https://maven.parchmentmc.org' }” a buildscript repositories:

```
build.gradle (forge-1.17.1) x  mods.toml x
1  buildscript {
2      repositories {
3          // These repositories are only for Gradle plugins, put any other repositories in the repository block further below
4          maven { url = 'https://maven.minecraftforge.net' }
5          maven { url = 'https://maven.parchmentmc.org' } //parchment mappings
6          mavenCentral()
7      }
8      dependencies {
9          classpath group: 'net.minecraftforge.gradle', name: 'ForgeGradle', version: '5.1.+', changing: true
10         classpath 'org.parchmentmc:librarian:1.+' //para el nuevo mapping
11     }
12 }
```

2. Añadir “classpath 'org.parchmentmc:librarian:1.+” en las dependencias, dentro de *buildscript*:

```
4      maven { url = 'https://maven.minecraftforge.net' }
5      maven { url = 'https://maven.parchmentmc.org' } //parchment mappings
6      mavenCentral()
7  }
8  dependencies {
9      classpath group: 'net.minecraftforge.gradle', name: 'ForgeGradle', version: '5.1.+', changing: true
10     classpath 'org.parchmentmc:librarian:1.+' //para el nuevo mapping
11 }
12 }
```

3. Añadir “apply plugin: 'org.parchmentmc.librarian.forgegradle” en la parte del código donde se encuentran las líneas que comienzan por “apply plugin: ”

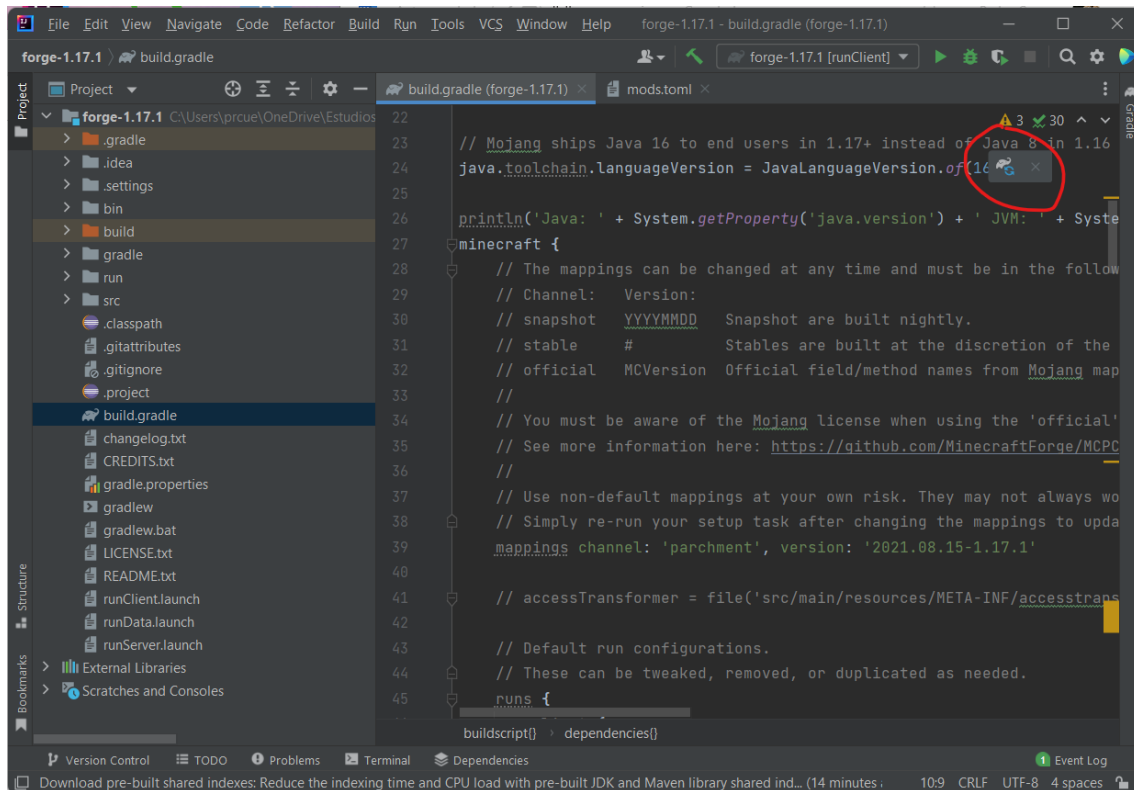
```
13  apply plugin: 'net.minecraftforge.gradle'
14  // Only edit below this line, the above code adds and enables the necessary things for Forge to be setup.
15  apply plugin: 'org.parchmentmc.librarian.forgegradle' //Nuevo mapping
16  apply plugin: 'eclipse'
17  apply plugin: 'maven-publish'
```



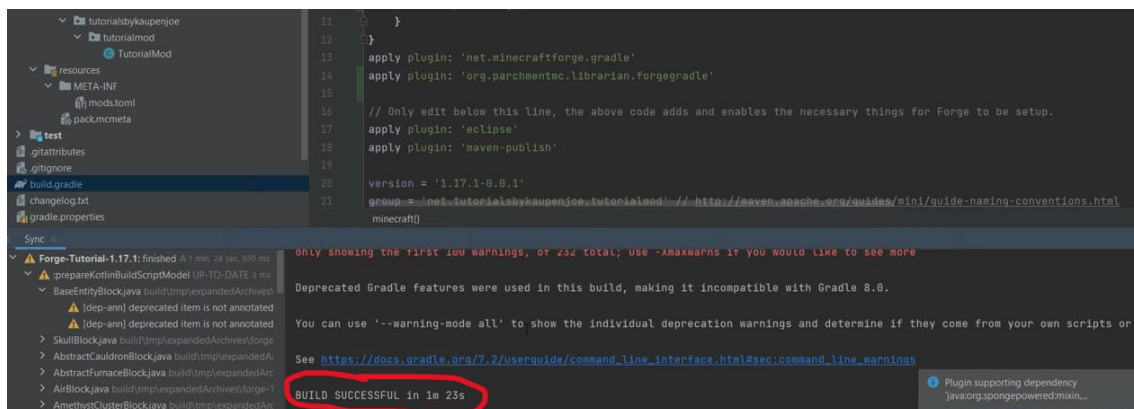
Proyecto ICAI - MINECRAFT



- Por último, haga clic en el botón “Load Gradle Changes”, que aparece en la esquina superior derecha de la pantalla:



Debemos esperar a recibir el mensaje de “BUILD SUCCESSFUL” en la parte inferior del programa:





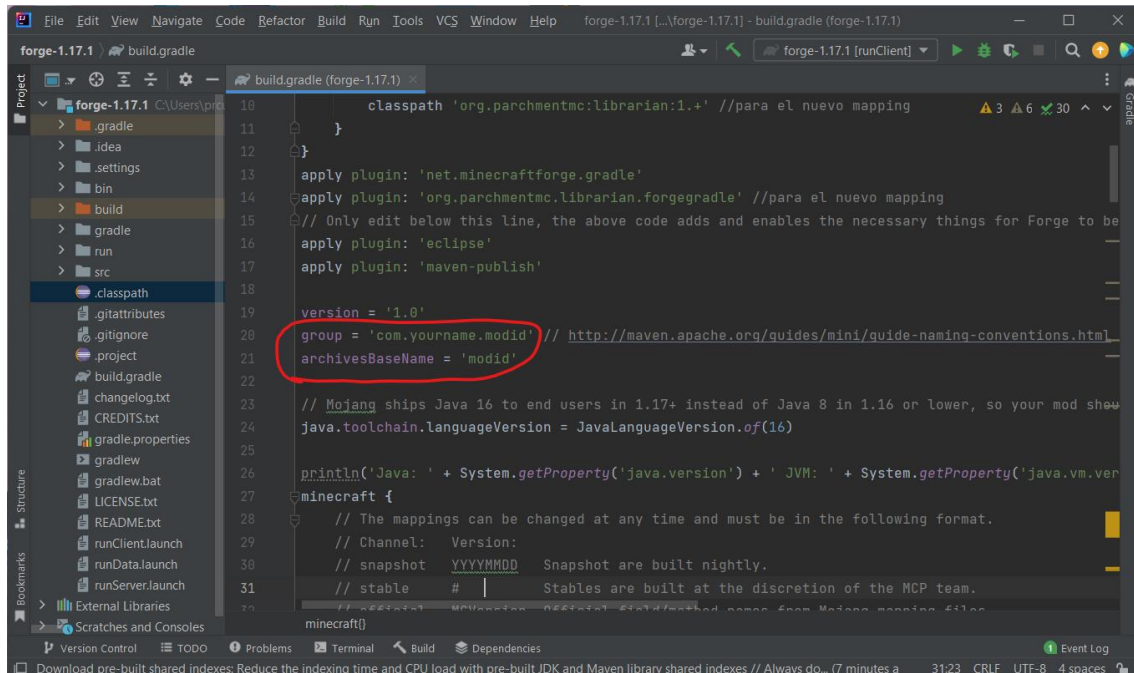
Proyecto ICAI - MINECRAFT



Creación del mod

Inicialización y personalización del mod

Sobre el mismo fichero de configuración (*build.gradle*) personalizaremos el nombre del mod. Comenzaremos otorgándole su propio *modid*, que es una cadena de caracteres que identifica a nuestro mod.



```
18 classpath 'org.parchmentmc:librarian:1.+' //para el nuevo mapping
19 }
20 }
21 apply plugin: 'net.minecraftforge.gradle'
22 apply plugin: 'org.parchmentmc.librarian.forgegradle' //para el nuevo mapping
23 // Only edit below this line, the above code adds and enables the necessary things for Forge to be
24 apply plugin: 'eclipse'
25 apply plugin: 'maven-publish'
26
27 version = '1.0'
28 group = 'com.yourname.modid' // http://maven.apache.org/guides/mini/guide-naming-conventions.html
29 archivesBaseName = 'modid'
30
31 // Mojang ships Java 16 to end users in 1.17+ instead of Java 8 in 1.16 or lower, so your mod should
32 java.toolchain.languageVersion = JavaLanguageVersion.of(16)
33
34 println('Java: ' + System.getProperty('java.version') + ' JVM: ' + System.getProperty('java.vm.version')
35
36 minecraft {
37     // The mappings can be changed at any time and must be in the following format.
38     // Channel:   Version:
39     // snapshot   YYYYMMDD   Snapshot are built nightly.
40     // stable     #           Stables are built at the discretion of the MCP team.
41     // Use 'auto' if you want to use the most recent snapshot.
42     mappings channel: 'snapshot', version: '20210803'
43
44     // Make an additional mapping to the source code, this is required for Forge
45     mappings {
46         src {
47             channel: 'snapshot', version: '20210803'
48         }
49     }
50 }
```

Debemos cambiar *yourname* y *modid* por los valores que prefiramos. El *modid* solo puede estar compuesto por minúsculas, números, guión bajo (_) y guión (-).

En este caso, los nombres serán:

```
19 version = '1.0'
20 group = 'com.comillas.mod_icai' // http://maven.
21 archivesBaseName = 'mod_icai'
```

A continuación, debemos usar el comando *replace* (Ctrl + R) para sustituir “*examplemod*” por nuestro *modid*, en nuestro caso, “*mod_icai*” en todo *build.gradle*:



```
61 mods {
62     examplemod {
63         source sourceSets.main
64     }
65 }
66
67 mods {
68     mod_icai {
69         source sourceSets.main
70     }
71 }
```

Ahora, volvemos a hacer clic sobre el botón de “Load Gradle Changes” que aparece en el extremo superior derecho de la pantalla. Debemos esperar a obtener el mensaje de “build successful” ya mencionado anteriormente:



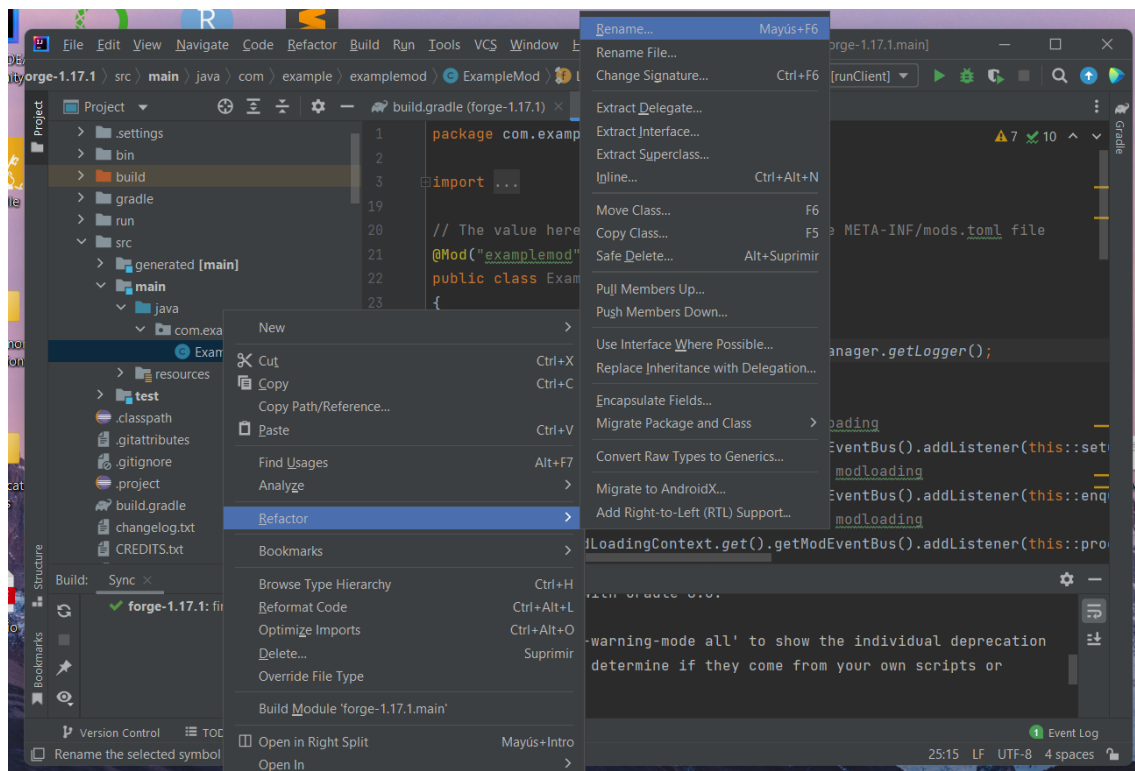
Proyecto ICAI - MINECRAFT



```
1 buildscript {
2     repositories {
3         // These repositories are only for Gradle plugins, put any other repositories
4         maven { url = 'https://maven.minecraftforge.net' }
5         maven { url = 'https://maven.parchmentmc.org' } //parchment mappings
6         mavenCentral()
7     }
8 }
9 dependencies {
10     classpath group: 'net.minecraftforge.gradle', name: 'ForgeGradle', version:
11     classpath 'org.parchmentmc:librarian:1.+ ' //para el nuevo mapping
12 }
13 apply plugin: 'net.minecraftforge.gradle'
14 apply plugin: 'org.parchmentmc.librarian.forgegradle' //para el nuevo mapping
15 // Only edit below this line, the above code adds and enables the necessary things
16 apply plugin: 'eclipse'
17 apply plugin: 'maven-publish'
18
19 version = '1.0'
20 jar {
21     manifest {
22     }
```

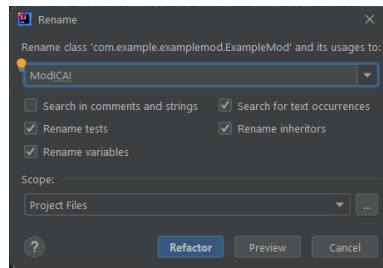
Modificación del nombre de la clase del Mod

A continuación, debe buscar la clase ExampleMod en src/main/java/[nombre del directorio escogido], hacer clic derecho sobre ella, y seleccionar refactor → rename para cambiar el nombre de la clase a **ModICAI** (por ejemplo).

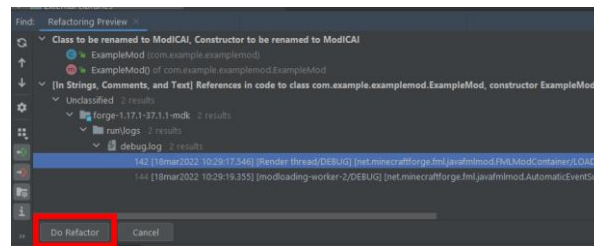




Proyecto ICAI - MINECRAFT



Una vez pulsado el botón *Refactor*, deberemos confirmar los cambios a realizar haciendo click en *Do Refactor*.





Proyecto ICAI - MINECRAFT



Modificación del código de la clase del Mod

Editar la clase *ModICAI* (cambiada de nombre) y eliminar todo el código existente desde la línea 45 hasta la línea 75, desde el método *enqueueIMC* hasta la clase estática *RegistryEvents*, ambos incluidos.

También se deberán eliminar las siguientes 4 líneas de código:

```
// The value here should match an entry in the META-INF/mods.toml file
@Mod("examplemod")
public class ModICAI {
    // Directly reference a log4j logger.
    private static final Logger LOGGER = LogManager.getLogger();

    public ModICAI() {
        // Register the setup method for modloading
        FMLJavaModLoadingContext.get().getModEventBus().addListener(this::setup);
        // Register the enqueueIMC method for modloading
        FMLJavaModLoadingContext.get().getModEventBus().addListener(this::enqueueIMC);
        // Register the processIMC method for modloading
        FMLJavaModLoadingContext.get().getModEventBus().addListener(this::processIMC);

        // Register ourselves for server and other game events we are interested in
        MinecraftForge.EVENT_BUS.register(this);
    }

    private void setup(final FMLCommonSetupEvent event)
    {
        // some preinit code
        LOGGER.info("HELLO FROM PREINIT");
        LOGGER.info("DIRT BLOCK >> {}", Blocks.DIRT.getRegistryName());
    }
}
```

A continuación, seleccionaremos el método ***getModEventBus ()*** de la línea de código “*FMLJavaModLoadingContext.get().getModEventBus().addListener(this::setup);*” y hacer clic derecho → Refactor → Introduce variable. Cambiad el nombre de la variable nueva por ***eventBus***.

```
public ModICAI() {
    // Register the setup method for modloading
    FMLJavaModLoadingContext.get().getModEventBus().addListener(this::setup);

    // Register ourselves for server and other game events we are interested in
    MinecraftForge.EVENT_BUS.register(this);
}

private void setup(final FMLCommonSetupEvent event)
{
    // some preinit code
}

// Register the setup method for modloading
// Register the enqueueIMC method for modloading
// Register the processIMC method for modloading
FMLJavaModLoadingContext.get().getModEventBus().addListener(this::setup);
FMLJavaModLoadingContext.get().getModEventBus().addListener(this::enqueueIMC);
FMLJavaModLoadingContext.get().getModEventBus().addListener(this::processIMC);
```

Modificación de la anotación @Mod

En el siguiente paso, debemos buscar “*@Mod("examplemod")*” en el inicio de la clase. Una vez lo encontremos, debemos seleccionar “examplemod”, comillas incluidas, y hacer clic derecho → refactor → Introduce Constant. Llamaremos a dicha constante ***MOD_ID*** y pulsaremos Enter. Con esto se estará creando un atributo de clase constante (final) que dará nombre al Mod mediante la anotación.

Dentro de la constante que se debería haber creado en la clase, debemos asignarle el valor de nuestro *modid*. Debería quedar de manera similar a la siguiente:



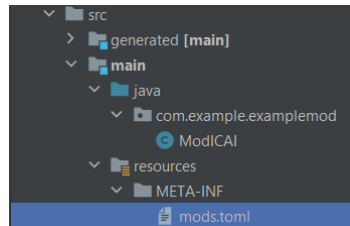
Proyecto ICAI - MINECRAFT



```
@Mod(ModICAI.MOD_ID)
public class ModICAI
{
    // Directly reference a log4j logger.
    private static final Logger LOGGER = LogManager.getLogger();
    public static final String MOD_ID = "mod_icaei";
}
```

Modificación del fichero mods.toml

Localizar el archivo *mods.toml* en la siguiente ubicación:

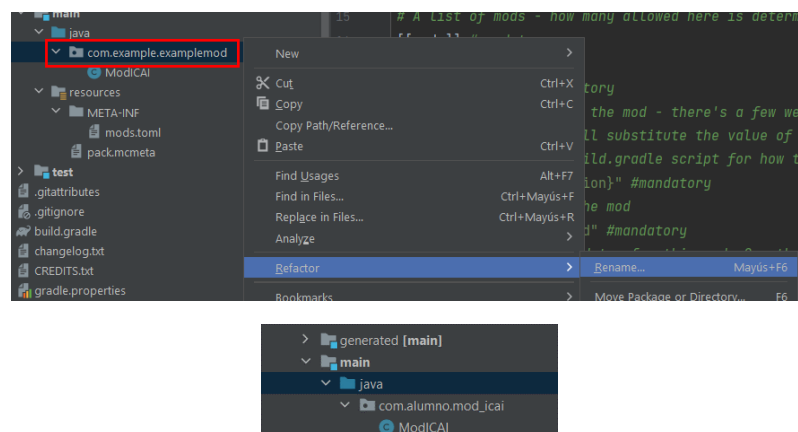


En dicho documento, deberíamos volver a actualizar el valor de *modid* con la opción de reemplazar nuevamente **Ctrl+R**. Buscar el texto **examplemod** y reemplazarlo por **mod_icaei**. Hacer clic en el botón *Replace All*. Además, cambiaremos también el *displayName* a “Mod de ICAI”.

```
15 # A list of mods - how many allowed here is determined by the mods.toml file
16 [[mods]] #mandatory
17 # The modid of the mod
18 modId="mod_icaei" #mandatory
19 # The version number of the mod - there's a few well known versions
20 # ${file.jarVersion} will substitute the value of the jarVersion in the mod
21 # see the associated build.gradle script for how to use the versioning system
22 version="${file.jarVersion}" #mandatory
23 # A display name for the mod
24 displayName="Mod de ICAI" #mandatory
```

Cambiar el nombre del paquete

Por último, deberíamos cambiar la estructura de los ficheros para que se corresponda con la que describimos en *build.gradle* (en nuestro caso era *com.comillas.mod_icaei*):

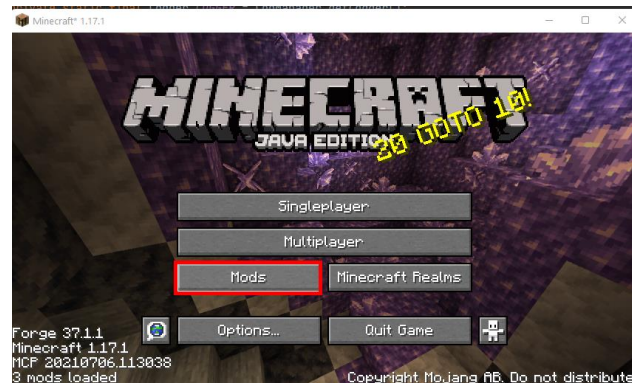


Si se ha quedado el paquete/carpeta *example* se puede borrar (teclar Supr.).

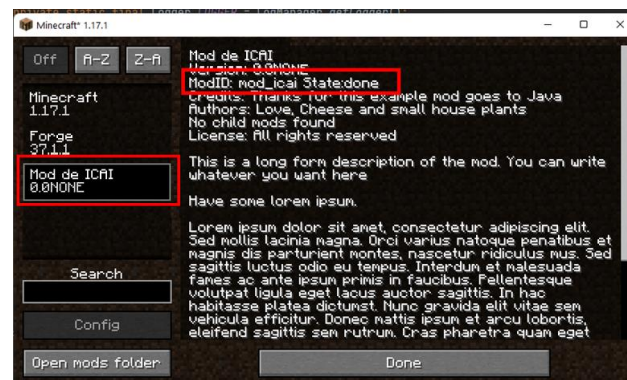
Deberíamos comprobar que todo funciona correctamente mediante la ejecución de Minecraft.



Proyecto ICAI - MINECRAFT



Nuestro mod debería aparecer en el menú de **Mods**:



Si quisiéramos modificar la información del Mod, bastaría con actualizar la información existente del fichero **mods.toml**. También se podría vincular un logo al mod.

