

# Developing

In this page you will find the requirements for developing FenixEdu applications. It begins by showing you what software you should install to begin, and provides some guidelines for choosing the best suited hardware for development.

## Software

### Operating System

FenixEdu applications run on any of the major operating systems (Linux, Mac OS and Windows) without any issue. However, the setup process on \*nix machines tends to be much easier.

You must ensure that your OS is 64-bit, or else some weird behavior might happen.

### Java

Most FenixEdu applications run on top of the Java platform. As such, to develop and run any application, you must have JDK 8 installed on your machine. You may choose between Oracle's JDK and OpenJDK, as long as they are on the latest version. Refer to your OS/distribution's documentation for the installation procedure.

The JDK's default heap size may be too low for development purposes. As such, you should set the `JAVA_OPTS` environment variable and define proper values. The recommended value is:

#### JAVA\_OPTS

```
-Xms1g -Xmx4g -Djava.awt.headless=true -XX:+UseG1GC -XX:+UseStringDeduplication
```

This will set the initial heap size to 1GB, maximum heap size to 4G, ensure the application runs in headless mode, and enable the G1 Garbage Collector.

### Maven

Maven is one of the build systems used to build FenixEdu applications. You need need to install Maven, usually the latest version will do. Currently, the latest version, 3.3.9, works fine.

You may install it using your OS's package manager, or downloading it from Maven's website.

Just like Java, the default Maven configuration may not be suitable for development. As such, you should set the `MAVEN_OPTS` environment variable, with the same value as `JAVA_OPTS`.

If you're running embedded Tomcat with Maven, you need to set the `MAVEN_OPTS` variable as such:

#### MAVEN\_OPTS

```
-Xms4g -Xmx8g -XX:MetaspaceSize=256m -verbose:gc -Djava.awt.headless=true -XX:+UseG1GC -XX:+UseStringDeduplication -Dorg.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING=false
```

This will set the initial heap size to 4GB, maximum heap size to 8GB, initial metaspace size to 256MB (to prevent full GC collections on startup), ensure the application runs in headless mode, enable the G1 Garbage Collector. The last flag is required for some legacy applications, see

[ACDM-472 - Review Strict Quote Escaping on JSPs](#) [OPEN](#) . You may want to tune the heap size values according to your computer's hardware configuration.

### MySQL/MariaDB

FenixEdu applications run on top of a MySQL or MariaDB database, using the InnoDB engine. As such, if you want to run a local database, you will have to install it on your machine.

If you want to run a "large" database (typically with multiple Gigabytes), you may need to tweak some configurations in your `my.cnf` file. The most important settings to tweak are the following:

## my.cnf

```
open_files_limit = 500000
max_connections      = 1000
max_allowed_packet  = 1G
thread_cache_size   = 128
default_storage_engine = InnoDB
innodb_buffer_pool_size = 20G
# Match this to the size in gigabytes of buffer_pool_size
innodb_buffer_pool_instances = 20
innodb_log_buffer_size = 128M
innodb_log_file_size = 1G
innodb_file_per_table = 1
innodb_open_files = 200000
innodb_io_capacity = 400
innodb_flush_method = O_DIRECT
innodb_autoinc_lock_mode = 2
innodb_flush_log_at_trx_commit = 0
innodb_read_io_threads = 32
innodb_write_io_threads = 32
innodb_io_capacity = 10000
innodb_fast_shutdown = 0
```

You will need to adjust the exact values according to your machine's hardware configuration. These values are for a dedicated machine with 8 cores and 32GB or RAM.

## Git

All FenixEdu projects are hosted in Git repositories. As such, the following contains a series of useful git aliases and configuration options you may want to use (try them yourself!):

## .gitconfig

```
[core]
    editor = emacs -nw

[alias]
    st = status

    # Show a colorful commit tree
    lsd = log --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%
an>%Creset' --abbrev-commit --date=relative
    lg = log --color=always -p
    dff = diff --color=auto
    dfc = diff --color=auto --cached
    fa = fetch --all -p

[help]
    autocorrect = 1

[grep]
    extendRegex = true
    lineNumber = true

[color]
    ui = true
```

If you want a nice Git prompt, you can find an example of a good one [here](#).

## Hardware

This section contains some guidelines regarding the best hardware choices for developing FenixEdu applications.

### CPU

Whereas the FenixEdu infrastructure is inherently multithreaded, rarely you will use multiple applications threads when developing. On the other hand, single-threaded performance is quite important for most tasks, such as compiling the code and running the application.

As such, it is recommended the usage of a CPU with a high clock speed and a modest core count. Recommended Intel Core i7 models (e.g. Intel Core i7 6700k).

## Memory

Being mostly developed in Java, memory is one of the most important aspects for your machine for developing FenixEdu applications.

The minimum RAM size is 8GB, and the recommended RAM size is 16GB. Currently, 32GB seems to be the RAM size that still delivers a good benefit for its price. ECC RAM is not necessary, and provides little to no benefit.

You should try to get the highest RAM speed for your platform, as it delivers great benefit in development environments.

## Disk

Your disk is one of the parts that impacts compilation and startup time performance. Using an SSD is a must, given that SSD prices are low enough to justify the performance benefits. HDDs are NOT RECOMMENDED.

PCI-e based SSDs provide the best performance possible, at competitive prices, and are thus the recommended platform. At the time of this writing, the recommended models are Samsung's 850 Pro for SATA, and Samsung's 950 Pro for PCI-e.

## Network

If you are working with a remote database, having a proper network configuration is critical. A low-latency single-hop gigabit connection is recommended. Usage of NICs with an Intel chipset provides the best performance results.