



PRACTICA 2

SISTEMA DE GESTIÓN DE ASCENSORES



Pablo Cabanyes Nieto

Esteban Navarro Melgarejo

Víctor Otero Hernández

Matías Sanz Ranz

Miguel Ángel Vara

Índice

Implementaciones y decisiones del código	2
Diagrama UML	4
Despliegue en Heroku	5
Reporte Sonarqube	6

Implementaciones y decisiones del código

Para el backend del sistema gestor de almacenes nos hemos decantado por la creación de dos clases principales: Ascensor y Edificio.

Debido a los requisitos que se nos pide en la práctica hemos decidido que la clase Ascensor debe seguir el patrón de diseño State y debe implementar observers. La decisión de usar un State fue al comprobar que se pedía que el ascensor pudiese estar en diferentes estados (moviéndose, abriendo puertas, cerrando puertas o parado). La decisión de implementar un observer fue debido a la petición en los requisitos de poder saber en qué piso se encontraban en todo momento los ascensores.

Clase Ascensor: Es la clase más importante del programa, cuenta con la variable **piso** que indica en que piso se encuentra el ascensor en todo momento (cuando se instancia un ascensor se indica el valor de esta variable como piso inicial en el que se encuentra). Cuenta también con **pisoObjetivo** y **pisoLlamada** que guardan el piso al que se quiere viajar y el piso desde el cual se ha llamado al ascensor respectivamente. Al utilizar el patrón de diseño State cuenta con una instancia de la interfaz Estado y con los métodos pertinentes de este patrón: setEstado(), abrirPuertas(), cerrarPuertas(), pararse(), también contamos con un método que nos indica en qué estado se encuentra el ascensor en todo momento saberEstado(). En esta clase hay también un método addObserver() para añadir observers y otro llamado removeObserver() para quitar observers, que son guardados en un array de observers y son utilizados mediante el método notifyObserver() que llama a update() y muestra la información de la forma en que ese observer la formatea.

Clase Edificio: En esta clase se realizan los viajes y peticiones de llamada de los ascensores. La clase anterior controlaba el funcionamiento de un ascensor, esta clase se encarga de su dirección y control. Cuenta con un ArrayList de la clase Ascensor que nos permite crear todos los ascensores que queramos (aunque luego solo vamos a usar 3). también cuenta con una clase llamarAscensor() y viajar(). llamarAscensor() se encarga de dirigir el ascensor al piso en el que ha sido llamado, creando un pequeño algoritmo que gestiona los diferentes estados del ascensor. El método viajar() se encarga de dirigir el ascensor al piso al que quiera dirigirse el usuario, controlando también los estados por los que tiene que pasar.

Como en el alcance no se contempla la optimización del movimiento de los ascensores estos no podrán pararse para recoger pasajeros si ya están metidos en un viaje.

Respecto al frontend realizado con vaadin: La pagina principal con 4 botones, 3 de ellos llamaran al ascensor que se desee y el cuarto mostrara una pestaña con el panel de control de los ascensores

Cuando se llama a un ascensor se tomará por defecto que el usuario se encuentra en el primer piso. Al llamar a un ascensor surgirá una nueva pantalla que contendrá la botonera de dicho ascensor, los 7 primeros botones indicaran al piso al que se quiere viajar, los 2 siguientes servirán para abrir y cerrar las puertas de la cabina, el siguiente mostrará en que piso se encuentra la cabina, el penúltimo mostrará el estado del ascensor(movimiento, abriendo puertas ...etc) y el ultimo será el botón de emergencia que mostrara la emisión del la emergencia.

El botón del panel de control mostrará 3 botones, que nos permitirán saber cual es el piso actual en el que se encuentra cada ascensor.

Toda la información que muestran los observers y el estado en el que se encuentra la cabina del ascensor serán mostrados por la consola de java.

PRÁCTICA 2 TAP

GESTIÓN DE ASCENSORES Y PANEL DE CONTROL

LLAMAR AL ASCENSOR 1

LLAMAR AL ASCENSOR 2

LLAMAR AL ASCENSOR 3

PANEL DE CONTROL

LLAMAR AL ASCENSOR 1

LLAMAR AL ASCENSOR 2

ASCENSOR 1

+ ×

ESTAS USANDO EL ASCENSOR 1

1

2

3

4

5

6

7

Abrir Puertas

Cerrar Puertas

Piso Actual

Estado de las puertas

EMERGENCIA

PANEL DE CONTROL

PANEL DE CONTROL

+ ×

Aquí se puede ver información de los ascensores

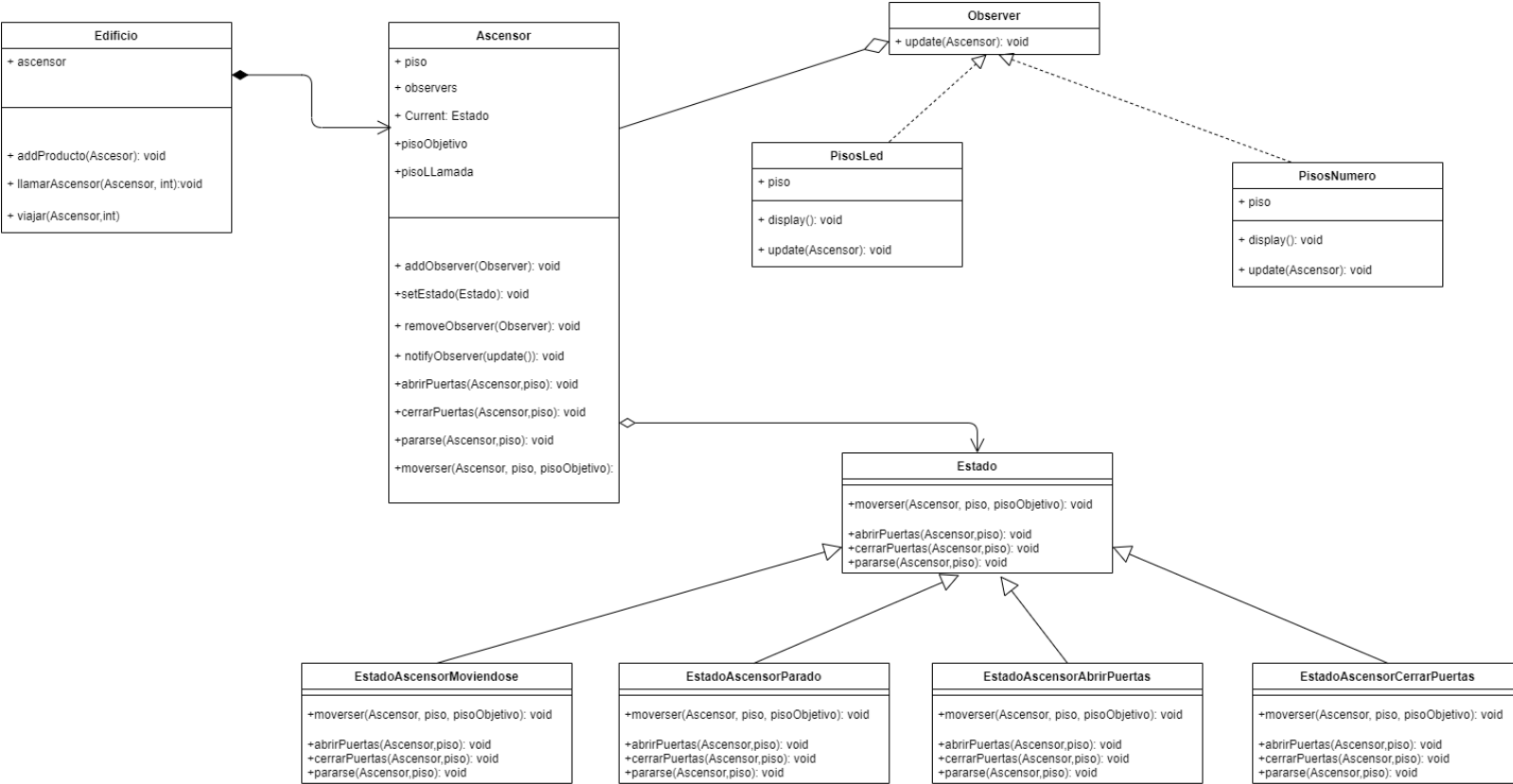
Piso en el que está el ascensor 1

Piso en el que está el ascensor 2

Piso en el que está el ascensor 3

EMERGENCIA

Diagrama UML



Despliegue en Heroku

Para poder ser accedida online hemos utilizado Heroku, donde la aplicación mantiene su estado sin contar el número de clientes conectado

Es importante destacar que hemos creado una nueva rama en Git para su despliegue, la página está en:

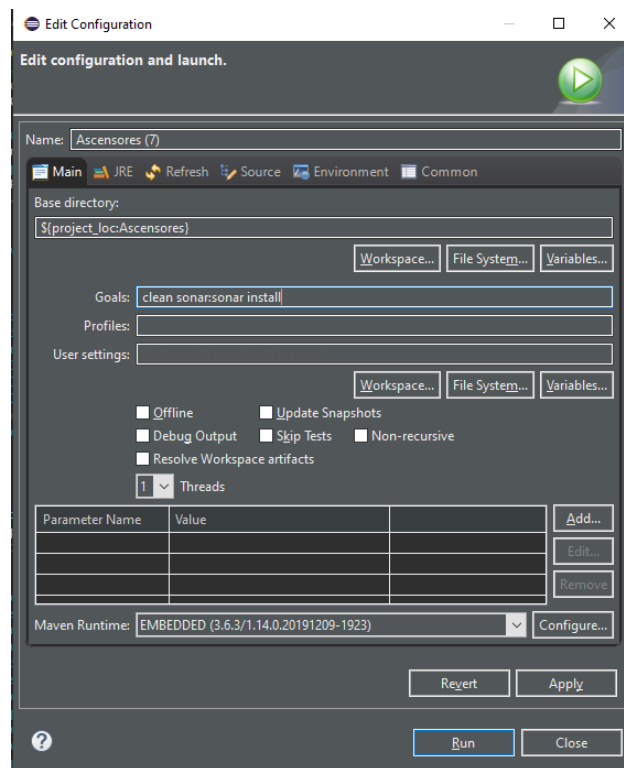
<https://ascensortap.herokuapp.com/>



Reporte Sonarqube

```
SonarQube
sers\altai\Downloads\sonarqube-8.6.0.39681\temp -XX:ErrorFile=../logs/es_hs_err_pid%p.log -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -XX:-OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -Dio.netty allocator.numDirectArenas=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Djava.locale.providers=COMPAT -Xmx512m -Xms512m -XX:MaxDirectMemorySize=256m -XX:+HeapDumpOnOutOfMemoryError -Delasticsearch -Des.path.home=C:\Users\altai\Downloads\sonarqube-8.6.0.39681\elasticsearch -Des.path.conf=C:\Users\altai\Downloads\sonarqube-8.6.0.39681\temp\conf\es -cp lib/* org.elasticsearch.bootstrap.Elasticsearch
jvm 1 | 2021.01.14 20:30:45 INFO app[[o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
jvm 1 | Java HotSpot(TM) 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
jvm 1 | 2021.01.14 20:30:53 INFO app[[o.s.a.SchedulerImpl] Process[es] is up
jvm 1 | 2021.01.14 20:30:53 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2, logFilenamePrefix=web]] from [C:\Users\altai\Downloads\sonarqube-8.6.0.39681]: C:\Program Files\Java\jdk-11.0.9\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\altai\Downloads\sonarqube-8.6.0.39681\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[:1] -cp ./lib/common/*;C:\Users\altai\Downloads\sonarqube-8.6.0.39681\lib\jdbc\h2\h2-1.4.199.jar org.sonar.server.app.WebServer C:\Users\altai\Downloads\sonarqube-8.6.0.39681\temp\sq-process3654988467213856711properties
jvm 1 | 2021.01.14 20:31:03 INFO app[[o.s.a.SchedulerImpl] Process[web] is up
jvm 1 | 2021.01.14 20:31:03 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFilenamePrefix=ce]] from [C:\Users\altai\Downloads\sonarqube-8.6.0.39681]: C:\Program Files\Java\jdk-11.0.9\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\altai\Downloads\sonarqube-8.6.0.39681\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[:1] -cp ./lib/common/*;C:\Users\altai\Downloads\sonarqube-8.6.0.39681\lib\jdbc\h2\h2-1.4.199.jar org.sonar.ce.app.CeServer C:\Users\altai\Downloads\sonarqube-8.6.0.39681\temp\sq-process7438430828444551168properties
jvm 1 | 2021.01.14 20:31:07 INFO app[[o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | 2021.01.14 20:31:07 INFO app[[o.s.a.SchedulerImpl] SonarQube is up
```

Nos hemos encontrado con problemas a la hora de llevar acabo el análisis con sonarqube. Primero hemos intentado hacerlo desde el propio eclipse, pero nos daba error en la version cuando teníamos instalada la última version.

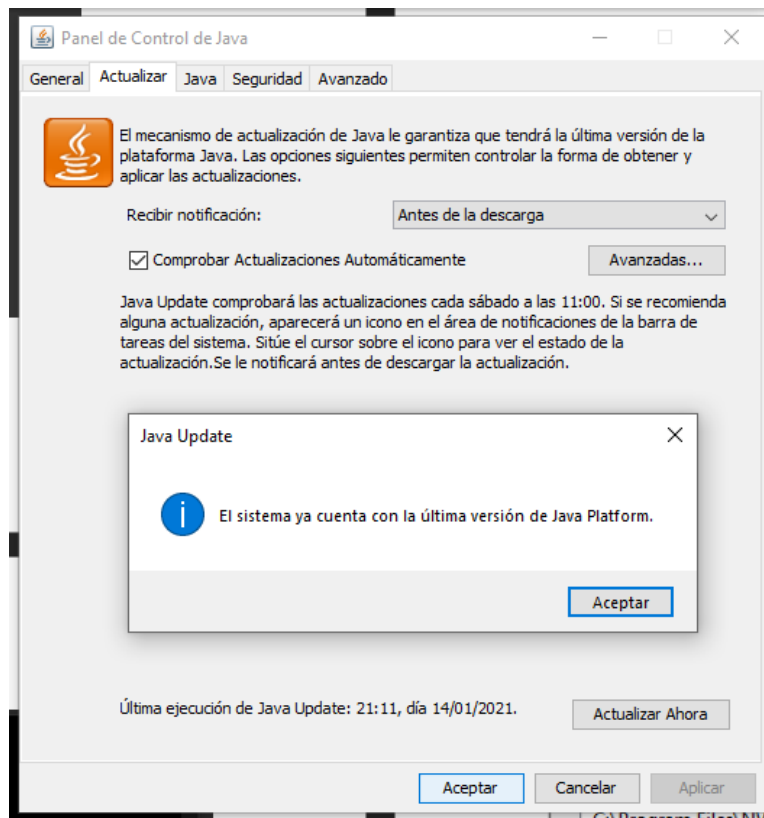


```

[INFO] -----< Tap:Ascensores >-----
[INFO] Building Ascensores 1.0-SNAPSHOT
[INFO] -----[ war ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.0.0:clean (default-clean) @ Ascensores ---
[INFO] Deleting C:\Users\altai\OneDrive\Escritorio\Gestion_ascensores-main\Ascensores\src\mai
[INFO]
[INFO] -----< Tap:Ascensores >-----
[INFO] Building Ascensores 1.0-SNAPSHOT
[INFO] -----[ war ]-----
[INFO]
[INFO] --- sonar-maven-plugin:3.8.0.2131:sonar (default-cli) @ Ascensores ---
[INFO] User cache: C:\Users\altai\.sonar\cache
[INFO] SonarQube version: 8.6.0
[INFO] Default locale: "es_ES", source code encoding: "UTF-8"
[WARNING] SonarScanner will require Java 11 to run starting in SonarQube 8.x
[INFO] Load global settings
[INFO]
[INFO] -----
[INFO] Skipping Ascensores
[INFO] This project has been banned from the build due to previous failures.
[INFO] -----
[INFO]
[INFO] BUILD FAILURE
[INFO] -----
[INFO] Total time: 2.185 s
[INFO] Finished at: 2021-01-14T20:31:09+01:00

```





Hemos probado reinstalando sonarqube y la version de java varias veces, quitando componentes de sonarqube que supuestamente provocaban este error y en todos los intentos nos saltaba el mismo fallo.

En otro de los intentos nos salto el fallo de version al reiniciar eclipse tras la instalacion de sonarqube y tuvimos que agregar manualmente en eclipse.ini la version que utilizaria y nos dejo arrancar eclipse.



```

-startup
plugins/org.eclipse.equinox.launcher_1.6.0.v20200915-1508.jar
--launcher.library
C:\Users\altai\.p2\pool\plugins\org.eclipse.equinox.launcher.win32.win32.x86_64_1.2.0.v20200915-1442
-product
org.eclipse.epp.package.java.product
-showsplash
C:\Users\altai\.p2\pool\plugins\org.eclipse.epp.package.common_4.18.0.20201210-1200
--launcher.defaultAction
openFile
--launcher.appendVmargs
-vm
C:\Program Files\Java\jdk-11.0.9\bin\java.exe
-vmargs
-Declipse.p2.max.threads=10
-Doomph.update.url=http://download.eclipse.org/oomph/updates/milestone/latest
-Doomph.redirection.index.redirection=index:/->http://git.eclipse.org/c/oomph/org.eclipse.oomph.git/plain/setups/
-Dosgi.requiredJavaVersion=11
-Dosgi.instance.area.default=@user.home/eclipse-workspace
-Dsun.java.command=Eclipse
-XX:+UseG1GC
-XX:+UseStringDeduplication
--add-modules=ALL-SYSTEM
-Dosgi.requiredJavaVersion=11
-Dosgi.dataAreaRequiresExplicitInit=true
-Xms256m
-Xmx2048m
--add-modules=ALL-SYSTEM

```

```

[INFO] Scanning for projects...
[WARNING] The project Tap:Ascensores:war:1.0-SNAPSHOT uses prerequisites which is only intended for maven-plugin projects but not for non maven-plugin projects. For such purposes you should use the maven
[INFO]
[INFO] -----< Tap:Ascensores >-----
[INFO] Building Ascensores 1.0-SNAPSHOT
[INFO] -----[ war ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.0.0:clean (default-clean) @ Ascensores ---
[INFO] Deleting C:\Users\altai\OneDrive\Escritorio\gestion_ascensores-main\Ascensores\src\main\webapp\VAADIN\themes (includes = [**/styles.css, **/styles.scss.cache], excludes = [])
[INFO]
[INFO] -----< Tap:Ascensores >-----
[INFO] Building Ascensores 1.0-SNAPSHOT
[INFO] -----[ war ]-----
[INFO]
[INFO] --- sonar-maven-plugin:3.8.0.2131:sonar (default-cli) @ Ascensores ---
[INFO] User cache: C:\Users\altai\.sonar\cache
[INFO] SonarQube version: 8.6.0
[INFO] Default locale: "es_ES", source code encoding: "UTF-8"
[INFO] Load global settings
[INFO]
[INFO]
[INFO] Skipping Ascensores
[INFO] This project has been banned from the build due to previous failures.
[INFO]
[INFO] BUILD FAILURE

```

Sin embargo nos seguia saliendo error al ejecutar pero ya no salia el error de la version.

Encontramos una forma de ejecutarlo desde el cmd de windows con el apache-maven, un sonarqube ejecutable de windows, estableciendo las variables de entorno requeridas y su referencia en la variable de entorno path. Sin embargo nos encontramos que no era capaz de reconocerlo el sistema.

```
Administrador: Símbolo del sistema
11 dirs 140.015.554.560 bytes libres

C:\>cd Users

C:\Users>dir
El volumen de la unidad C es Windows
El número de serie del volumen es: 4255-7E17

Directorio de C:\Users
13/06/2020 23:44 <DIR> .
13/06/2020 23:44 <DIR> ..
14/01/2021 20:25 <DIR> altai
14/08/2020 00:20 <DIR> Public
0 archivos 0 bytes
4 dirs 140.016.672.768 bytes libres

C:\Users>cd altai
C:\Users\altai>cd OneDrive
C:\Users\altai\OneDrive>cd Escritorio
C:\Users\altai\OneDrive\Escritorio>cd Gestion_ascensores-main
C:\Users\altai\OneDrive\Escritorio\Gestion_ascensores-main>mvn sonar:sonar
"mvn" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
C:\Users\altai\OneDrive\Escritorio\Gestion_ascensores-main>
```

Variables de entorno

Variables de usuario para altai

Variable	Valor
OneDrive	C:\Users\altai\OneDrive
OneDriveConsumer	C:\Users\altai\OneDrive
Path	C:\Users\altai\AppData\Local\Microsoft\WindowsApps;C:\Users\altai\AppData\Local\GitHubDesktop\bin;C:\Program Files\heroku\bin;C:\Users\altai\AppData\Local\Temp
TEMP	C:\Users\altai\AppData\Local\Temp
TMP	C:\Users\altai\AppData\Local\Temp

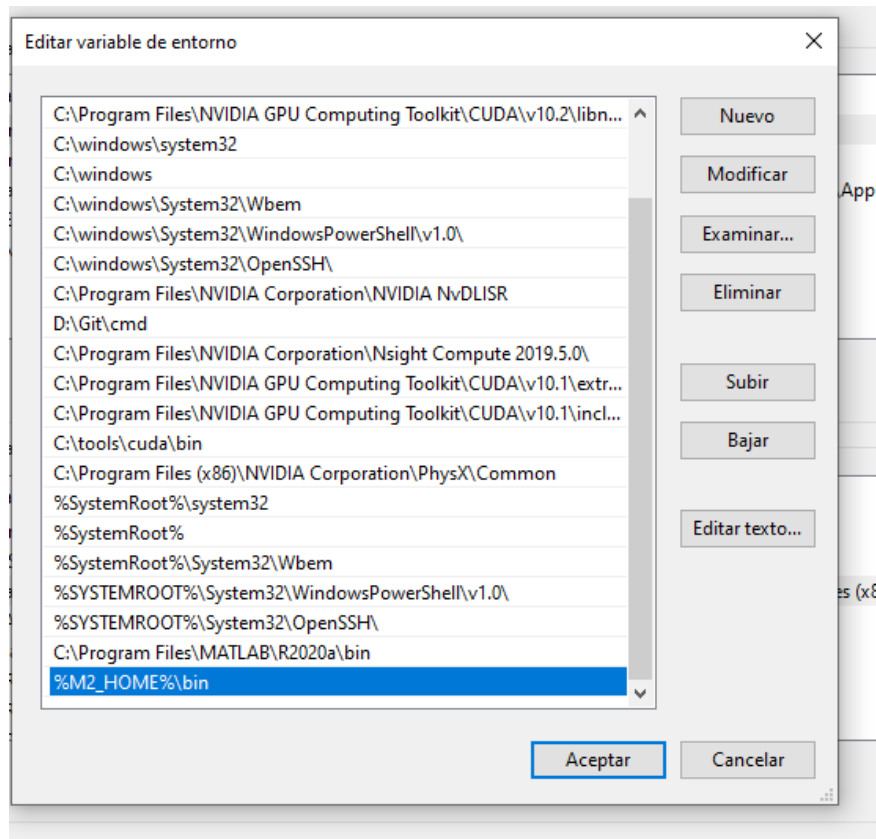
Nueva... Editar... Eliminar

Variables del sistema

Variable	Valor
ComSpec	C:\WINDOWS\system32\cmd.exe
CUDA_PATH	C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2
CUDA_PATH_V10_2	C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2
DriverData	C:\Windows\System32\Drivers\DriverData
M2_HOME	C:\Users\altai\Downloads\apache-maven-3.6.3
NUMBER_OF_PROCESSORS	8
NVCUDASAMPLES_ROOT	C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2
NVCUDASAMPLES_V10_2_ROOT	C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2

Nueva... Editar... Eliminar

Aceptar Cancelar



Hemos probado distintas formas de declarar las variables de entorno, distintos paquetes de apache-maven, usar el cmd como administrador y no nos reconocía el comando mvn de ninguna de las formas.