



Certificada en: Diseño,
desarrollo, implantación y
soporte de sistemas
informáticos,
de comunicaciones y
automatización.

CECOGEST

Documentació per a desenvolupadors

CONTROL DOCUMENTACIÓ

Descripció del document

En aquest document s'inclou informació que pot resultar d'interès per a desenvolupadors.

Històric de versions

VERSIÓ	RESPONSABLE	DATA	DESCRIPCIÓ
1.0	Limit Technologies	27/10/2018	Versió original
1.1	Limit Technologies	30/10/2019	Modificació estructura

Índex

1. Configuració d'un entorn de desenvolupament per Cecocloud.....	4
1. Instal·lació de l'IDE.....	4
2. Lombok.....	5
3. Obtenció del codi font.....	6
4. Execució de l'aplicació.....	11

1. Configuració d'un entorn de desenvolupament per Cecocloud

1. Instal·lació de l'IDE

L'aplicació recomanada per a les tasques de desenvolupament de l'aplicació és Eclipse en la seva darrera versió. En els exemples d'aquest document s'utilitza la versió és **2019-09**.

L'IDE es pot descarregar de la pàgina de descàrregues del projecte:

<https://www.eclipse.org/downloads/>

Des d'aquesta pàgina es pot accedir a les diferents versions d'eclipse (Download Packages)



Aquesta pàgina ofereix diferents paquets configurats per diferents tasques i llenguatges de programació. El paquet que recomanem descarregar és «Eclipse IDE for Enterprise Java Developers».



Per a instal·lar l'aplicació basta descomprimir-la a una carpeta. S'ha de tenir en compte que, per a funcionar l'aplicació, es necessita tenir instal·lat una versió del JDK de Java. Per a Eclipse **2019-09** es requereix una versió Java 8 o superior.

2. Lombok

El projecte utilitza la llibreria lombok (<https://projectlombok.org/>). Aquesta llibreria serveix per a generar de forma automàtica cert codi repetitiu com els setters, getters, constructors, hashCode, equals, ...

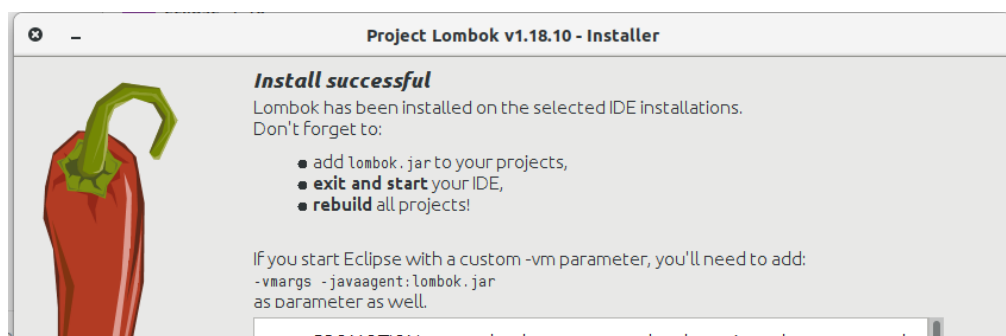
Per a utilitzar aquesta llibreria, és necessari que s'instal·li al IDE que utilitzem.

Per instal·lar la llibreria lombok seguirem les següents passes:

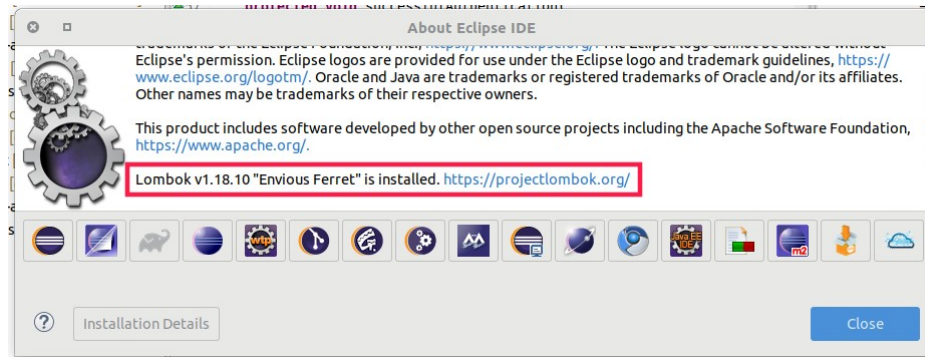
1. Descarregar el fitxer lombok.jar (<https://projectlombok.org/downloads/lombok.jar>)
2. Executar el fitxer lombok.jar: `java -jar lombok.jar`
3. Seleccionar la carpeta de l'eclipse



4. Selecciona la opció Install / Update



5. Un cop reiniciat l'IDE, es pot comprovar que el lombok està instal·lat anant a Help > About Eclipse IDE, i al final del text de copyright es mostrarà la versió de lombok

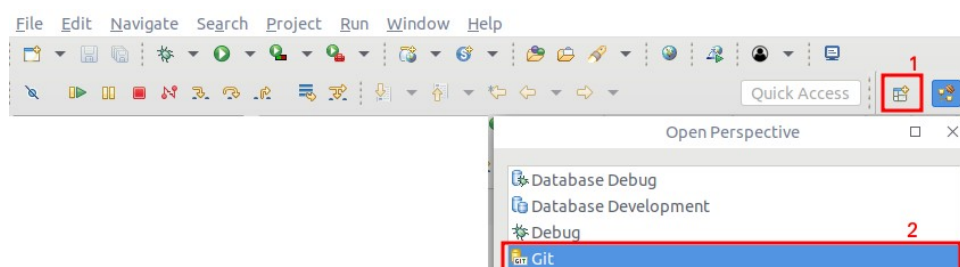


3. Obtenció del codi font

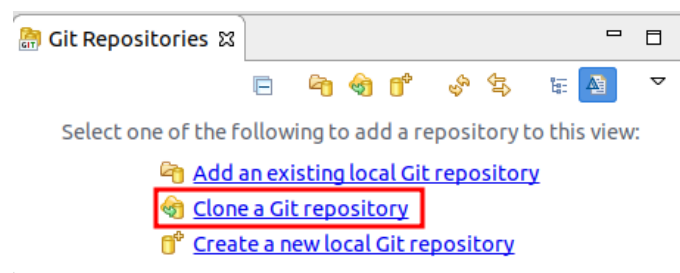
El codi font del projecte es troba allotjat en repositoris privats de GitHub. Per a poder accedir a aquests repositoris és necessari posar-se en contacte amb el personal de Limit, que s'encarregaran de donar permisos sobre aquests.

Un cop es disposi d'un usuari de GitHub amb permisos, es podrà utilitzar el mateix IDE Eclipse per descarregar i configurar el codi font del projecte seguint les següents passes:

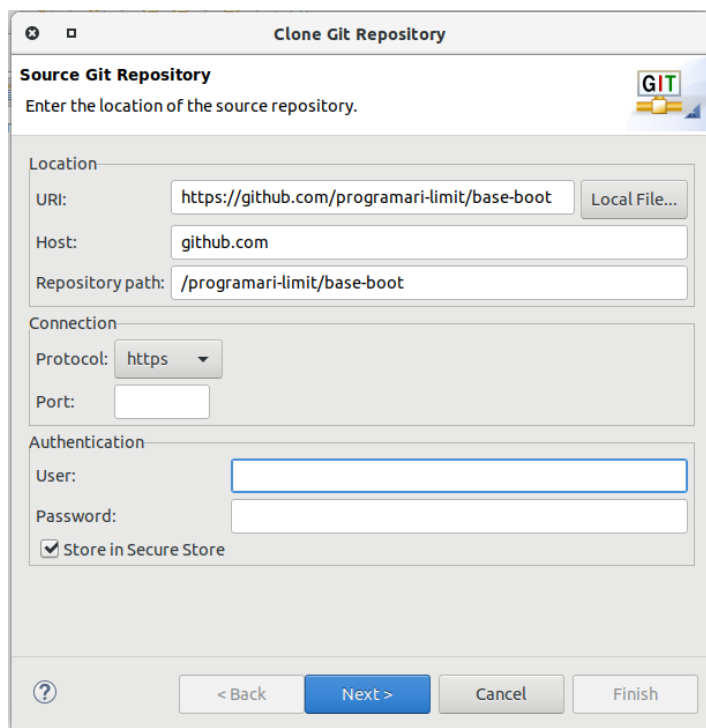
1. He d'obrir la perspectiva Git d'Eclipse pitgant al boto "Open perspective" i seleccionam la opció "Git".



2. Des de aquesta perspectiva hem de activar la opció "Clone a Git repository"

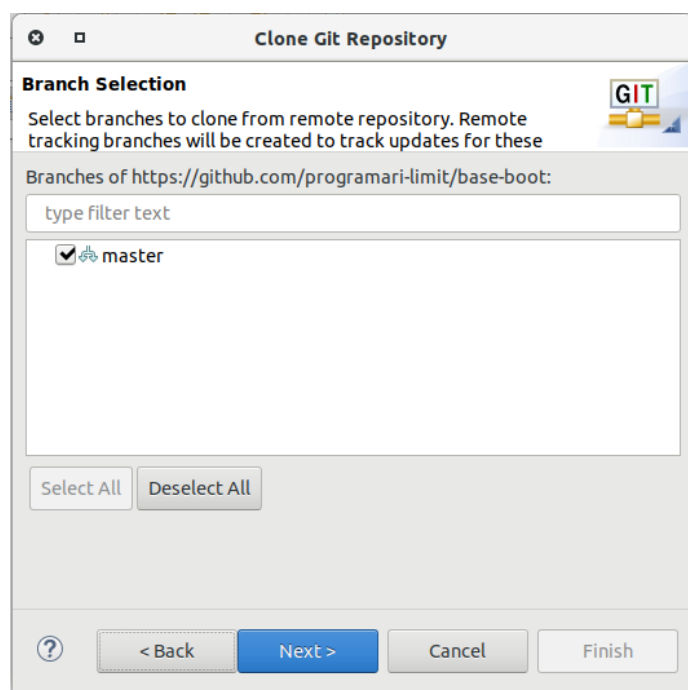


3. A la següent finestra hem d'inserir al camp "URI" la direcció del nostre repositori remot i hem d'emplenar les nostres dades d'autenticació.



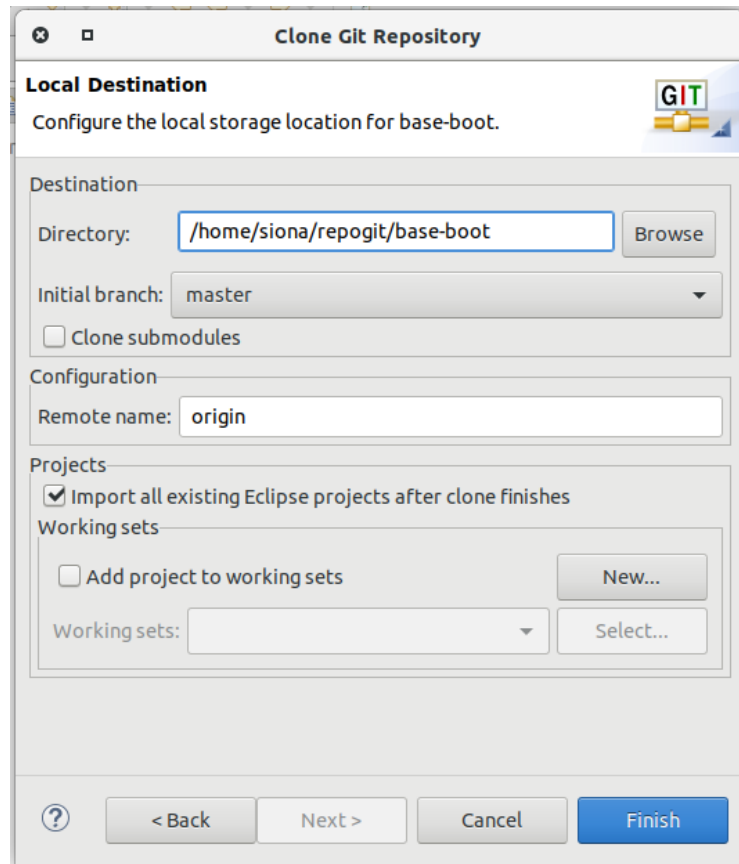
The dialog box is titled "Clone Git Repository". It has a "Source Git Repository" section with the instruction "Enter the location of the source repository." and a "GIT" logo. The "Location" section contains three fields: "URI:" with the value "https://github.com/programari-limit/base-boot", "Host:" with "github.com", and "Repository path:" with "/programari-limit/base-boot". There is a "Local File..." button next to the URI field. The "Connection" section has a "Protocol:" dropdown set to "https" and an empty "Port:" field. The "Authentication" section has "User:" and "Password:" fields, and a checked checkbox for "Store in Secure Store". At the bottom are buttons for "< Back", "Next >", "Cancel", and "Finish".

4. Triam les branques que volem davallar

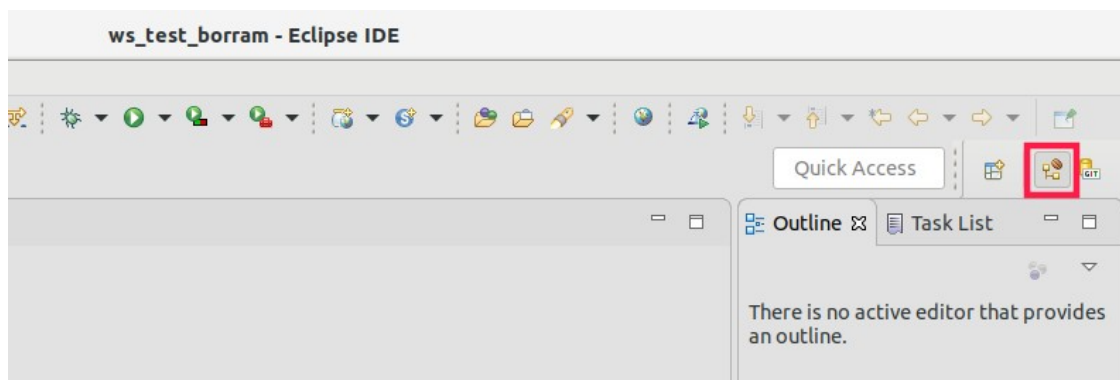


The dialog box is titled "Clone Git Repository". It has a "Branch Selection" section with the instruction "Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these" and a "GIT" logo. Below this, it says "Branches of https://github.com/programari-limit/base-boot:". There is a "type filter text" input field. A list box contains one item: "master" with a checked checkbox and a branch icon. At the bottom of the list box are "Select All" and "Deselect All" buttons. At the bottom of the dialog are buttons for "?", "< Back", "Next >", "Cancel", and "Finish".

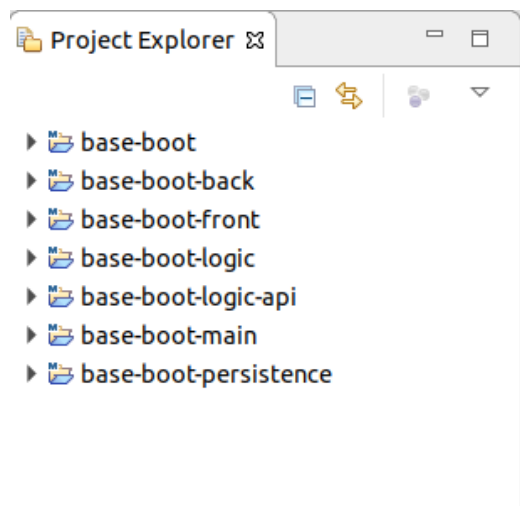
5. Finalment indicam a quin directori s'ha de clonar el repositori a la nostra maquina i marcam l'opció "Import all existing Eclipse projects after clone finishes".



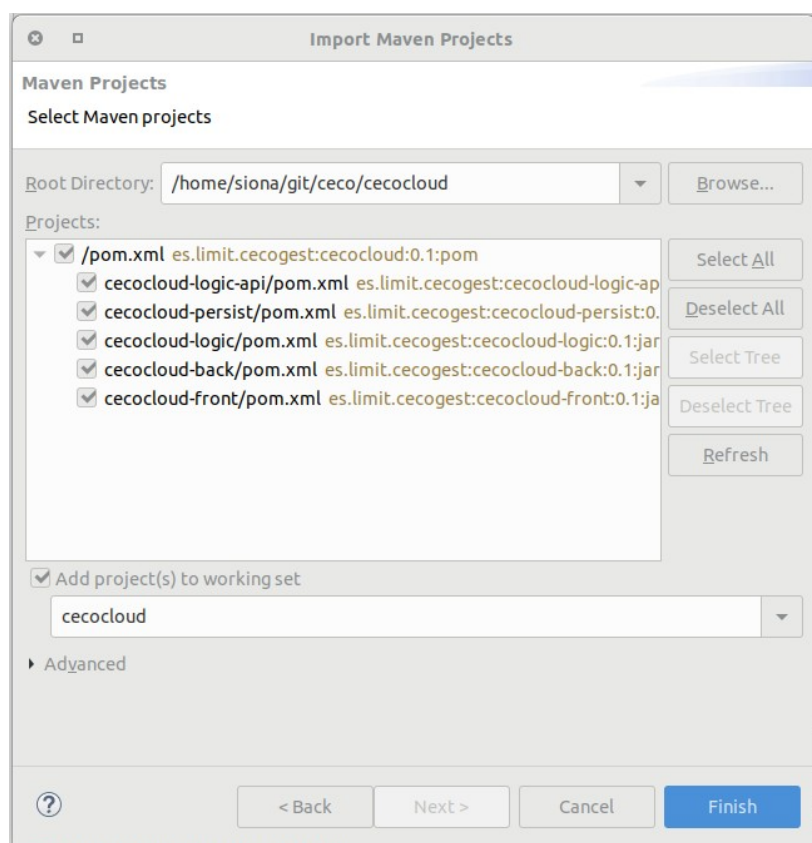
Una vegada finalitzat el procés de descarrega tornam a la perspectiva "Java EE"



i podrem veure l'estructura del projecte.



En cas que eclipse no importi el projecte, es pot fer manualment. Seleccionant la opció File > import..., indicant Maven > Existing Maven Projects, i finalment seleccionar la carpeta on hem desat el repositori git.



Serà necessari descarregar 2 repositoris:

- <https://github.com/programari-limit/base-boot>
- <https://github.com/programari-limit/cecocloud>

4. Base de dades

Crear una nova base de dades amb PostgreSQL i inicialitzar amb els següents scripts:

1. cecocloud-main/src/main/resources/schema-postgresql.sql
2. cecocloud-main/src/main/resources/data-postgresql.sql

5. Configuració de l'aplicació

Crear el fitxer de propietats application.properties directament dins cecocloud-main, amb les següents propietats:

```
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQL9Dialect
spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation=true
spring.datasource.url=jdbc:postgresql://localhost:5432/[DBNAME]
spring.datasource.username=[DBUSER]
spring.datasource.password=[DBPASS]
```

```
spring.mail.host=correu.limit.es
spring.mail.port=465
spring.mail.username=proves_limit@limit.es
spring.mail.password=uyw77RED
spring.mail.properties.mail.smtp.from=cecocloud@limit.es
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.ssl.enable=true
spring.mail.properties.mail.smtp.ssl.trust=correu.limit.es
spring.mail.properties.mail.smtp.starttls.enable=true
```

* S'ha de substituir [DBNAME], [DBUSER] i [DBPASS] amb els valors definits en la instal·lació de la base de dades.

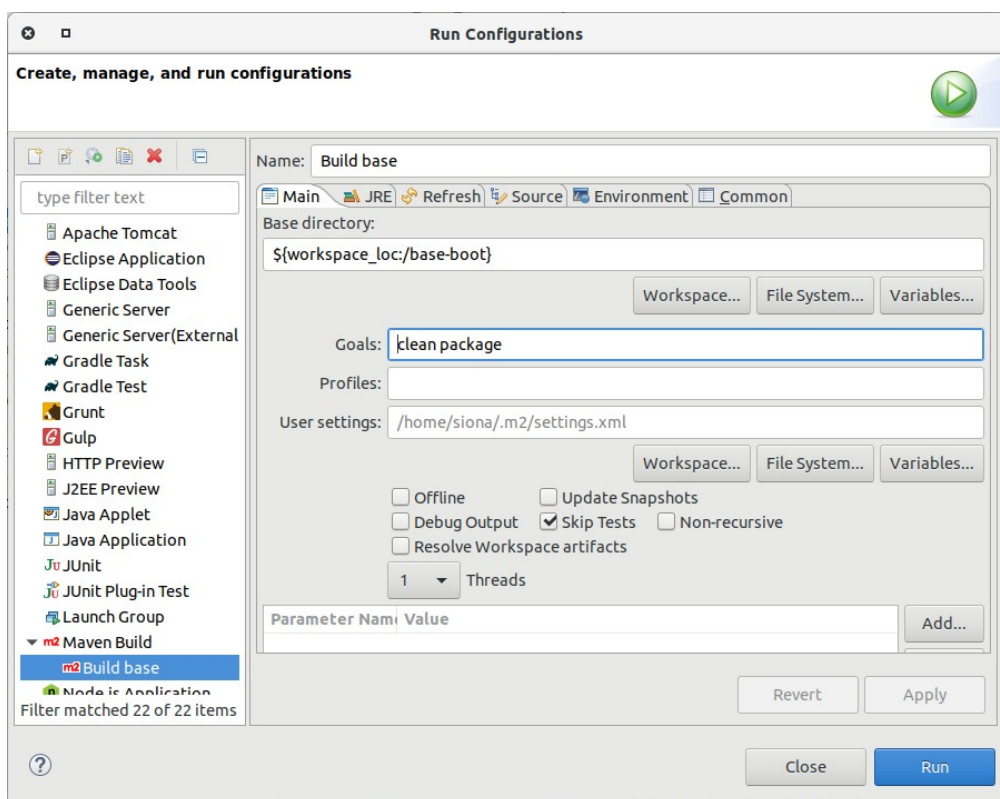
6. Execució de l'aplicació

Per poder executar **Cecocloud** per primera vegada hem de seguir les següents passes:

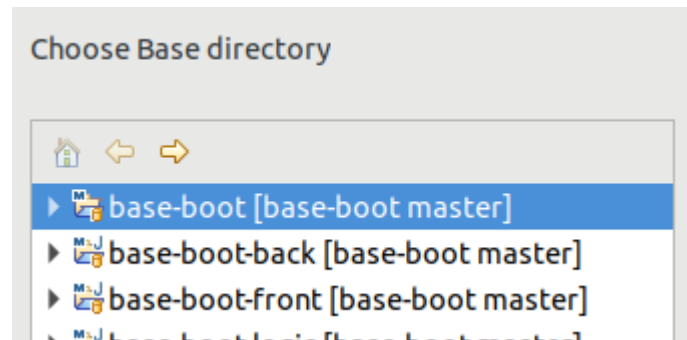
1. Dins del explorador del projecte, feim clic amb el botó dret damunt qualsevol carpeta del projecte base-boot i seleccionam la opció “Run As > Run Configurations”.
2. En la següent finestra hem de cercar la “Maven Build” i hi feim clic amb el botó dret i seleccionam la opció “New Configuration”.

m2 Maven Build

3. Es obrira la següent pantalla on hem de emplenar el camp “Base directory”, el camp “Goals” i hem de marcar “Skip Tests”.



- **Base directory:** Hem de pitjar damunt el botó “Workspace” i seleccionam la carpeta “cecogest”.

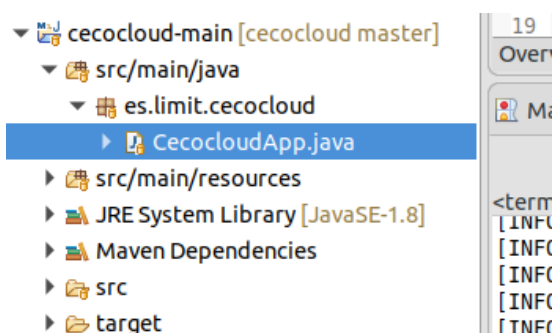


- **Goals:** clean package
 - **Skip Tests:** Aquest camp es marca per que ignori els tests.
4. Finalment pitjem al botó “Apply” per guardar la configuració i després “Run” per executar la instal·lació.

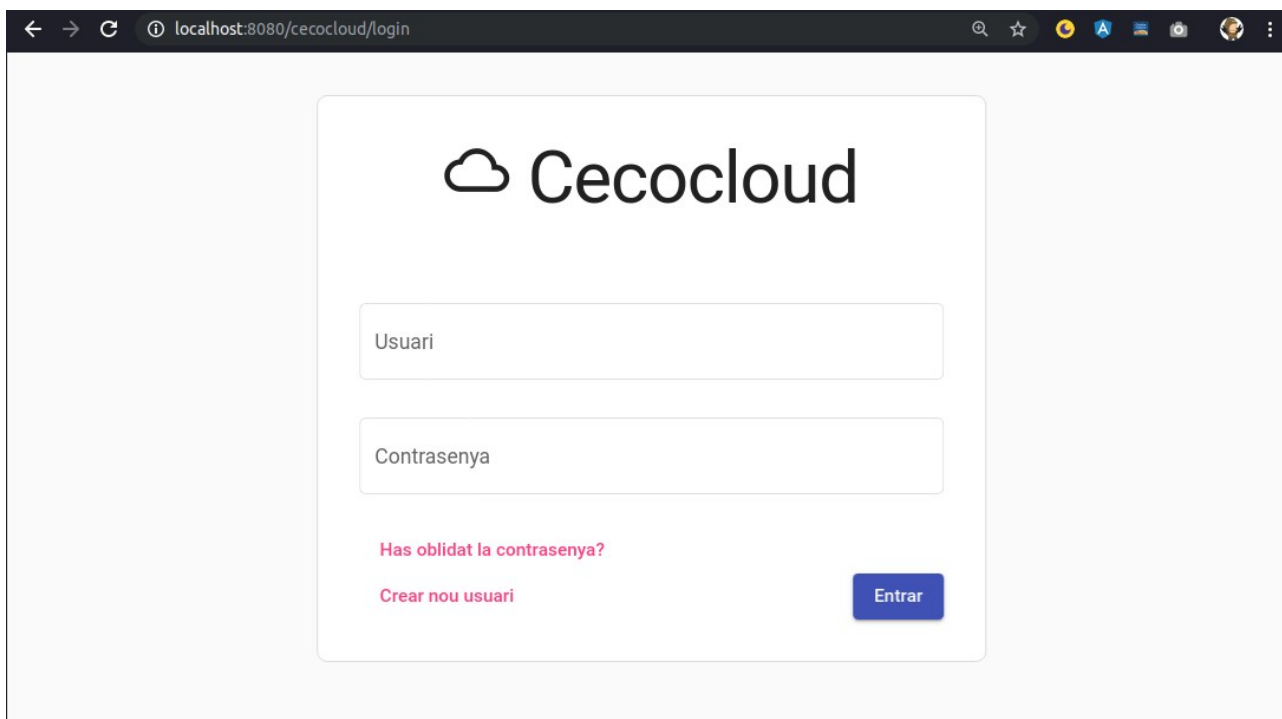
```
[INFO] --- spring-boot-maven-plugin:2.1.6.RELEASE:repackage (repackage) @ base-boot-main ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] Reactor Summary for base-boot 0.1:
[INFO]
[INFO] base-boot ..... SUCCESS [ 0.135 s]
[INFO] base-boot-logic-api ..... SUCCESS [ 2.017 s]
[INFO] base-boot-persistence ..... SUCCESS [ 0.574 s]
[INFO] base-boot-logic ..... SUCCESS [ 0.980 s]
[INFO] base-boot-back ..... SUCCESS [ 0.667 s]
[INFO] base-boot-front ..... SUCCESS [ 1.206 s]
[INFO] base-boot-main ..... SUCCESS [ 0.717 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.579 s
[INFO] Finished at: 2019-10-30T12:07:10+01:00
[INFO] -----
```

5. Instal·larem els paquets de base-boot al nostre repositori maven local. Per fer això, dins del explorador del projecte, feim clic amb el botó dret damunt la carpeta base-boot i seleccionam la opció “Run As > Maven Install”
6. Configurar les credencials per a accedir als components npm del base-boot. Per això executarem des de línia de consola la comanda:
- ```
npm login --registry=https://npm.pkg.github.com
```
- **Username:** usuari de github
  - **Password:** TOKEN que el personal de Limit hagi proporcionat
  - **Email:** el correu electrònic associat al compte de github.
7. Repetirem les passes de la 1 a la 4 per al projecte cecocloud:

- **Base directory:** Hem de pitjar damunt el botó “Workspace” i seleccionam la carpeta “cecogest”.
  - **Goals:** clean package
  - **Skip Tests:** Aquest camp es marca per que ignori els tests.
8. Al finalitzar la instal·lació del projecte ‘Maven’ ens hem de dirigir a la classe “**cecocloud-main/src/main/java/es.limit.cecocloud/CecocloudApp.java**”, hi feim clic dret a damunt i seleccionam “Run As > Java Application”.



9. Una vegada iniciada l’aplicació ens podem dirigir al nostre navegador i introduïm la següent direcció: <http://localhost:8080/cecocloud>.



Per a desenvolupar el frontend (Angular) modificarem la configuració del projecte per tal d'utilitzar l'entorn de desenvolupament i no entorn de producció.

Els canvis a realitzar són els següents:

1. Posar la propietat **skip.npm**, definida al fitxer cecocloud/pom.xml, a **true**.
2. Anar a la carpeta cecocloud/cecocloud-fron/src/main/resources i executar:  
./npm install
3. Cada cop que s'executi l'aplicació executarem en aquesta mateixa carpeta:  
./ng build --watch