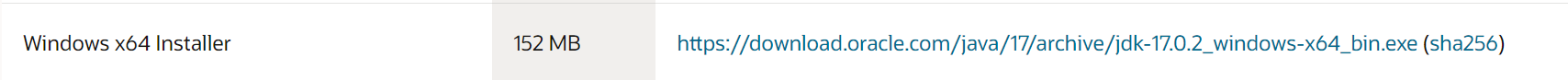
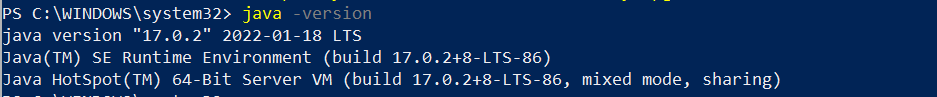
**PRÁCTICA 1 PROGRAMACIÓN DE APLICACIONES TELEMÁTICAS**

En primer lugar, descargamos los programas necesarios para el desarrollo de la asignatura:

* Java 17





* Maven

Me he descargado maven desde PowerShell una vez tenía Chocolatey con el comando choco install maven:



Texto

Descripción generada automáticamente

* Intellig

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Un letrero de color blanco

Descripción generada automáticamente con confianza baja

Captura de pantalla de un celular

Descripción generada automáticamente

Una vez he descargado Java17, Maven e IntelliJ, paso a la prueba de los comandos:

* Para clonar un repositorio, como por ejemplo hello world para esta práctica, es necesario el siguiente comando:



* Una vez ya tenemos el repositorio clonado, podemos ver el estado del repositorio. En este caso, he añadido un fichero de prueba para probar cómo subirlo. Como podemos ver en la foto siguiente, este fichero no está al día.

Texto

Descripción generada automáticamente

* Para poder añadir un documento al repositorio personal, es necesario hacer git add en primer lugar, como para preparar. Este comando es necesario además para guardar los cambios una vez se haya modificado algún documento; sin embargo, hay que tener en cuenta de que este documento no será visible en el repositotio.

Texto

Descripción generada automáticamente

* Una vez ya tenemos los cambios guardados, procedemos a subir todo al repositorio, para que sea visible al público.

Para ello, es necesario hacer un commit en primer lugar y un push en segundo lugar.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

La diferencia entre estos dos comandos es que commit es para subir el documento al repositorio local y push sirve para que sea visible a cualquiera que quiera consultar el repositorio remoto.

Una vez ejecutados todos estos comandos, debería estar subido el documento de prueba al repositorio, lo comprobamos:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Por último, probaré el comando git checkout

Texto

Descripción generada automáticamente

Lo que hemos hecho en este caso es, en un primer lugar, crear una nueva Branch llamada feature/1 y cambiarnos a ella para trabajar en esa Branch.

En segundo lugar, he vuelto a la Branch que estábamos usando inicialmente para así seguir trabajando en el directorio donde tengo todo.

Para terminar la práctica, subiré este Word a mi repositorio de GitHub para su correcta evaluación.