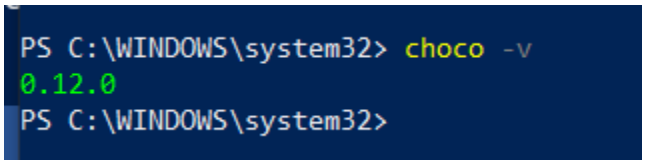
PRACTICA 1: Plataforma Github & Git

Mencía Sánchez Rosillo

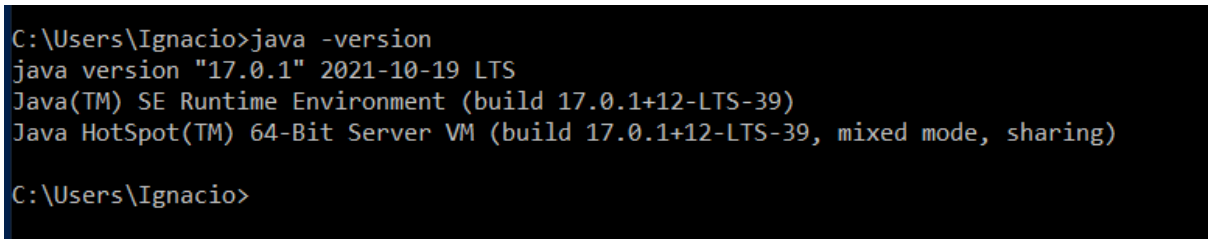
3 GITT+BA

Programación de Aplicaciones Telemáticas

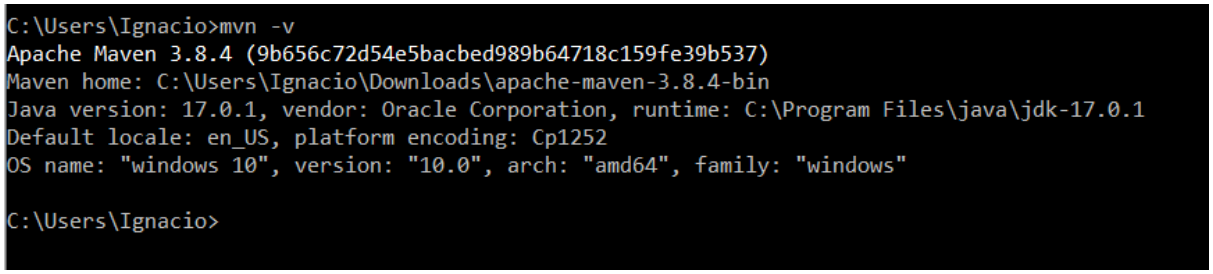
En la primera practica de Programación de Aplicaciones Telemáticas, empezamos descargándonos las aplicaciones y los servicios necesarios para la asignatura. En primer lugar, nos instalamos la plataforma "Chocolatey" que trata de una herramienta para descargar diferentes aplicaciones en Windows desde un amplio repositorio, utilizando tan solo el terminal.

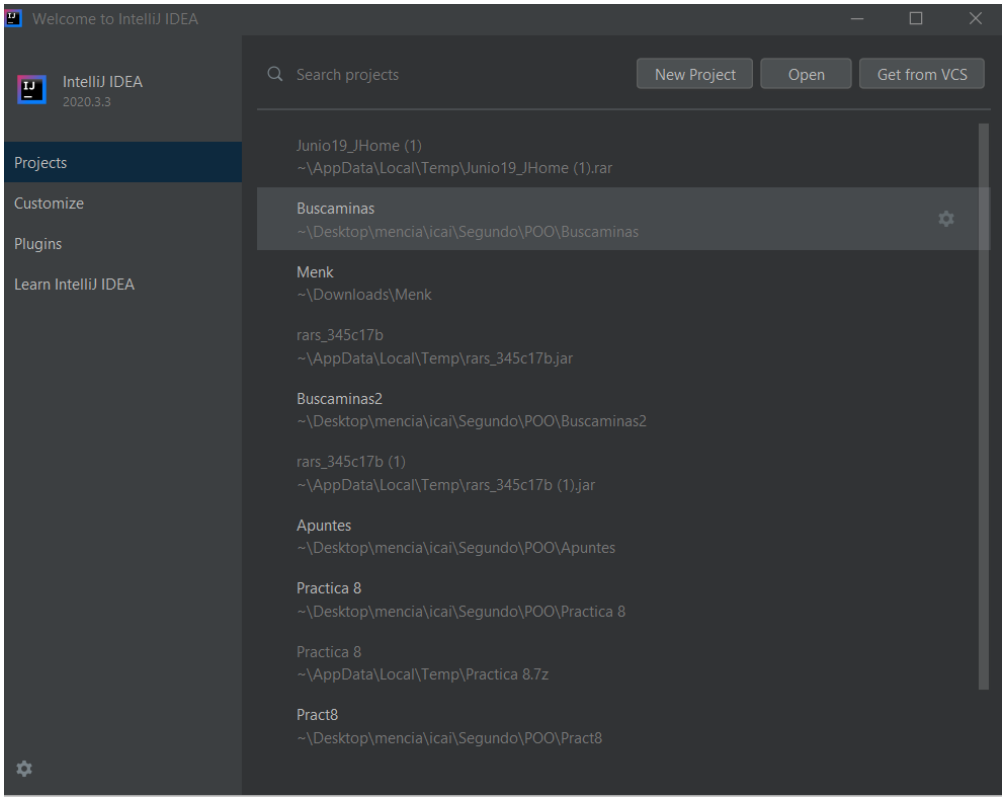


A continuación, comprobé la versión de java que tenia descargada de otras asignaturas, pero no era la 17. Por lo tanto, procedí a descargarme la versión correcta y comprobar que estaba bien descargada.



Mas adelante, me descargue Maven y comprobé que seguía teniendo IntelliJ de asignaturas previas.





El siguiente paso, fue comprobar que tenia una cuenta de Github y con mi cuenta de usuario hice un fork sobre el repositorio: https://github.com/gitt-3-pat/hello-world. Una vez tengo el repositorio en mi cuenta (https://github.com/201900831/hello-world) abro un entorno de Gitpod; gitpod.io/#https://github.com/ALUMNO-GITT-PAT/hello-world, y pruebo los siguientes comandos.

Git clone – Este comando sirve para fijar como objetivo un repositorio que ya existe con el objetivo de copiarlo (o como el nombre indica de "clonarlo"). Ahora mismo hemos copiado, el repositorio en el que estoy dentro de mi repositorio "hello-world". Sirve para copiar repositorios.

```
gitpod /workspace/hello-world $ git clone https://github.com/gitt-3-pat/hello-world
Cloning into 'hello-world'...
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 38 (delta 1), reused 31 (delta 0), pack-reused 0
Receiving objects: 100% (38/38), 58.97 KiB | 7.37 MiB/s, done.
Resolving deltas: 100% (1/1), done.
gitpod /workspace/hello-world $
```

Git status- Este comando te mostrara los diferentes estados de los archivos en tu directorio de trabajo. Te va diciendo que archivos están modificados y sin seguimiento y cuales con seguimiento pero que no están confirmados aún. Si esta actualizado o no.

```
gitpod /workspace/hello-world $ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello-world/

nothing added to commit but untracked files present (use "git add" to track)
```

Git add . – Se usa para agregar archivos al área de preparación. En este caso, el punto sirve para todo, es guardar todos los cambios que hayas hecho en github. El punto significa todos los archivos.

```
gitpod /workspace/hello-world $ git add .
warning: adding embedded git repository: hello-world
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> hello-world
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached hello-world
hint:
hint: See "git help submodule" for more information.
gitpod /workspace/hello-world $
```

Gitt commit - guardará todos los cambios hechos en la zona de montaje o área de preparación, junto con una breve descripción del usuario, en un "commit" al repositorio local. Una característica importante es que puedes recordar los cambios a los que se les hizo commits en una fecha posterior.

```
gitpod /workspace/hello-world $ git commit -m "TU MENSAJE"
[main b1c7663] TU MENSAJE
 1 file changed, 1 insertion(+)
 create mode 160000 hello-world
gitpod /workspace/hello-world $
```

Gitt push- es un comando que sube los cambios hechos en tu ambiente de trabajo a una rama de trabajo tuya. Commit identifica los cambios hechos en dicho ambiente de trabajo. Si tu no haces un push de tus cambios, estos jamás se verán reflejados en tu repositorio remoto.

Me salía un error al principio, pero después de darle los permisos todo correcto.

```
gitpod /workspace/hello-world $ git push
remote: Permission to 201900831/hello-world.git denied to 201900831.
fatal: unable to access 'https://github.com/201900831/hello-world.git/': The requested URL returned error: 403
```

```
gitpod /workspace/hello-world $ git push
Everything up-to-date
```

Git checkout -b feature/1 – Es un comando que sirve para desplazarte entre las diferentes ramas o restaura los archivos del árbol de trabajo. En este caso, como se puede ver en el mensaje dice que se ha cambiado a una nueva rama “Feature/1” que es la que le hemos dicho nosotros. Este sirve para crear y hacer checkout a una rama nueva con un solo comando.

```
gitpod /workspace/hello-world $ git checkout -b feature/1
Switched to a new branch 'feature/1'
gitpod /workspace/hello-world $
```

Git checkout main- El git checkout es como he explicado en el apartado anterior, pero en este caso estamos realizando un checkout a una rama existente “main”.

```
gitpod /workspace/hello-world $ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
gitpod /workspace/hello-world $
```