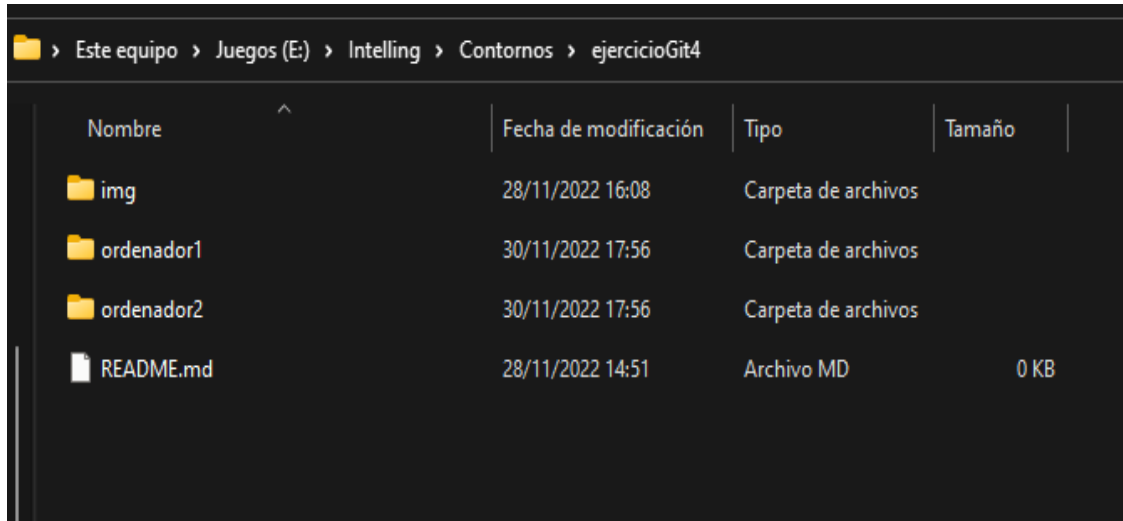


<https://github.com/andvigofp/ejercicioGit4> – Mi repositorio, para poder seguir la guía,

1- Creamos una carpeta de trabajo con:

- Dos carpetas (Ordenador1 y Ordenador2), que simularán dos equipos remotos.
- Este README.md para ir editándolo y creando el registro de trabajo con capturas.
- Una carpeta "img" para las imágenes que serán referenciadas desde el README



2- Creamos un repositorio vacío (sin README ni gitignore) en GitHub.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * andvigofp / Repository name * ejercicioGit4 ✓

Great repository names are short and memorable. Need inspiration? How about [turbo-octo-lamp](#)?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▼

Choose a license

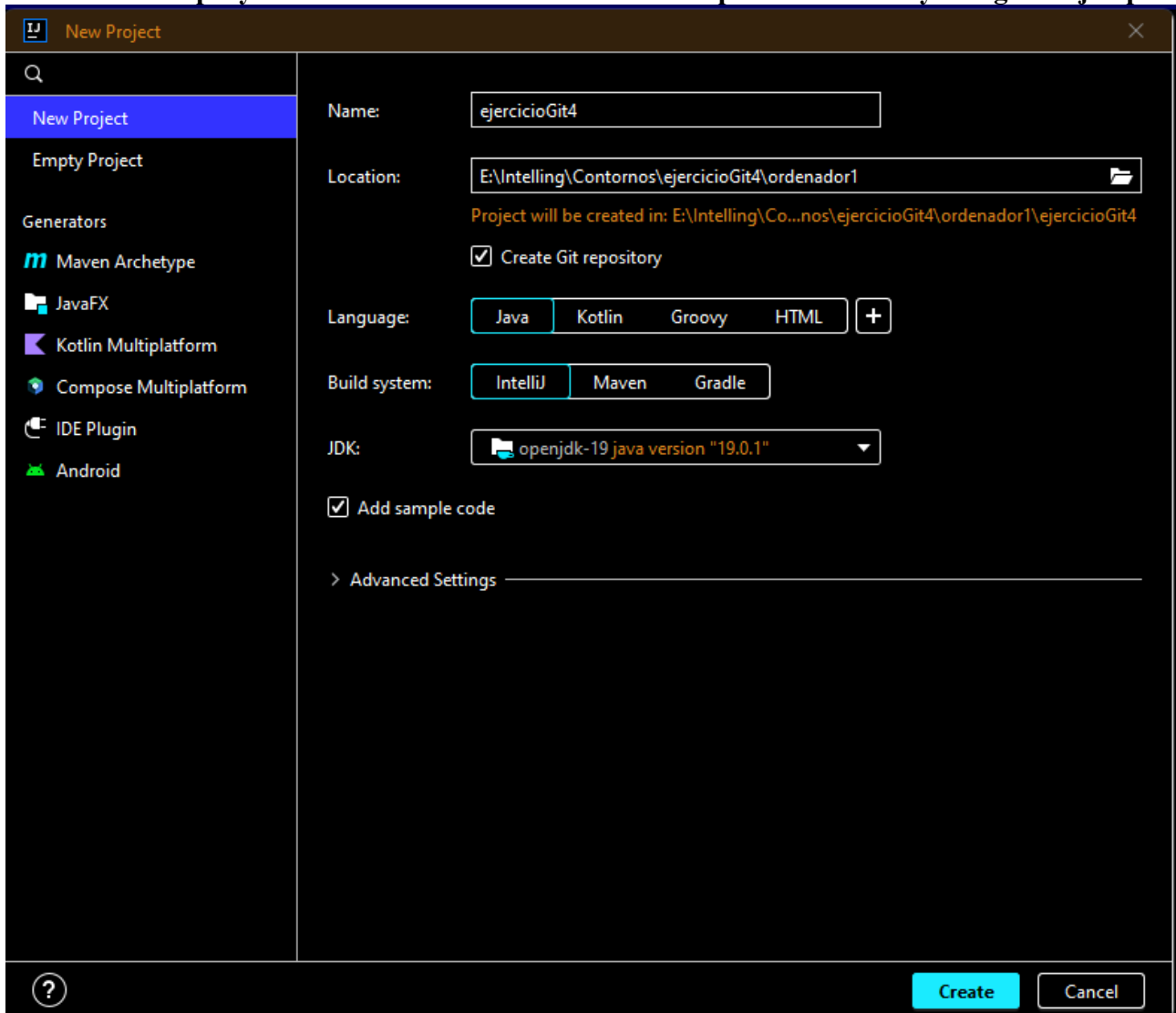
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▼

ⓘ You are creating a public repository in your personal account.

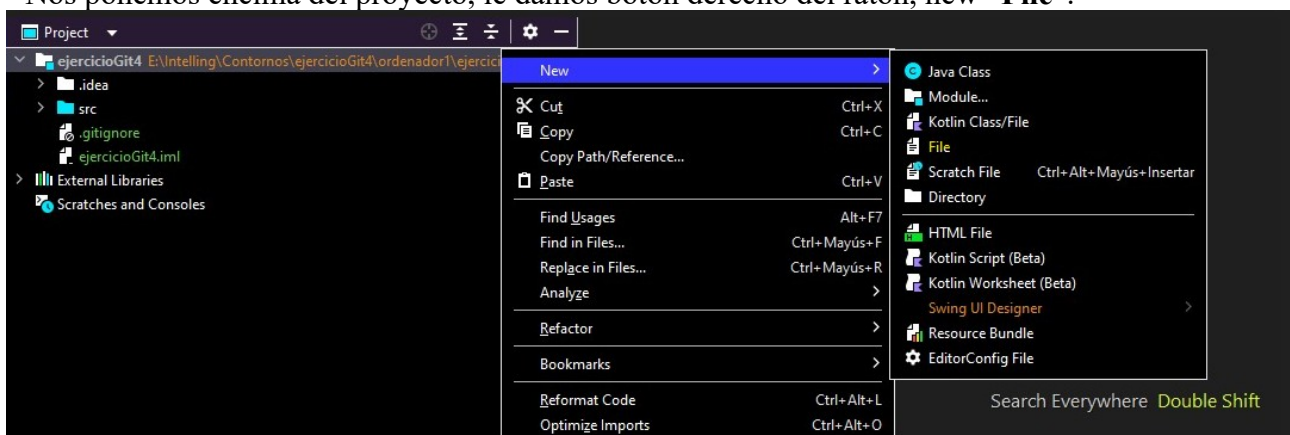
Create repository

3- Creamos un proyecto de IntelliJ en Ordenador1 con repositorio de Git y código de ejemplo.



4- Creamos un nuevo README.md localmente (no el presente con el registro de trabajo, que se añadirá al final sustituyendo a ese.)

- Nos ponemos encima del proyecto, le damos botón derecho del ratón, new “File”.



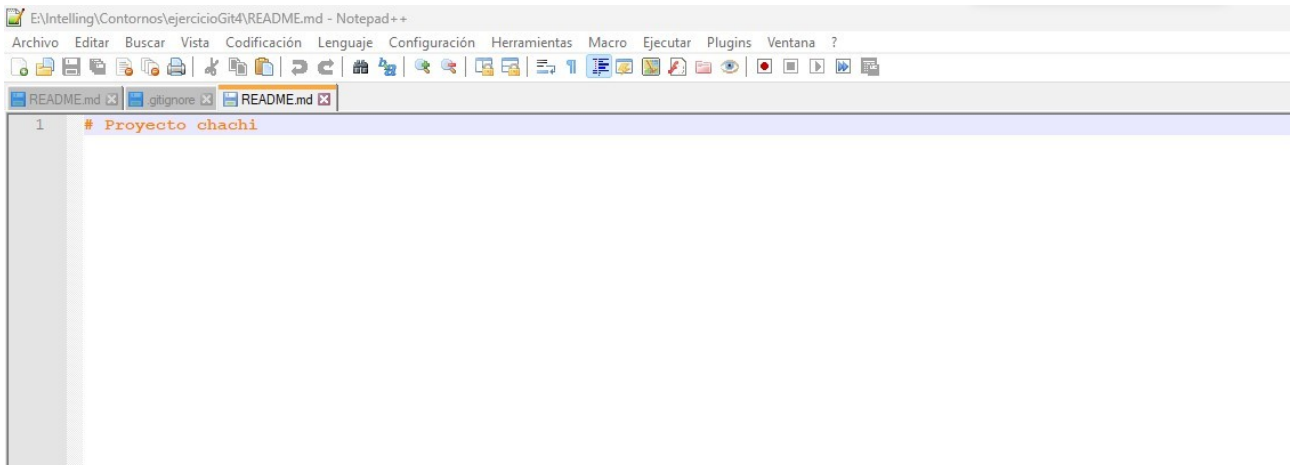
- Nos saldrá este cuadro, ponemos “**README.md**” (para crear el archivo **README.md**)



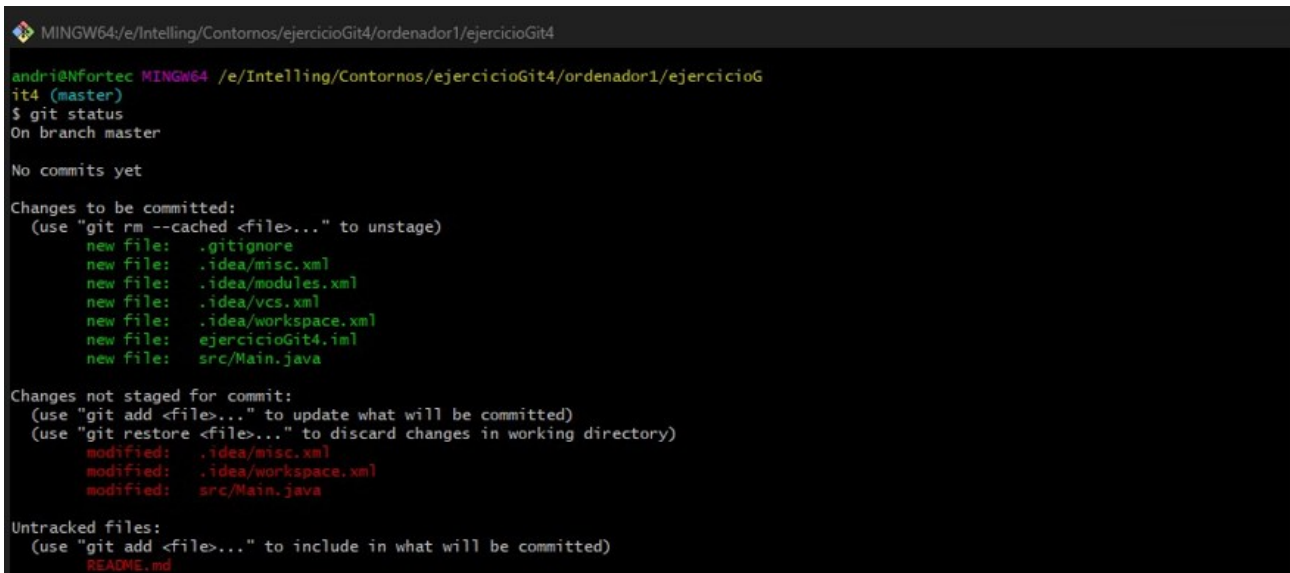
5 -Realizamos un primer commit y push de la base del proyecto.

NOTA: ved mi log para comprobar que hacéis commits idénticos a los míos.

- Editamos el archivo **README.md**



- Comprobamos el estado de los archivos antes de preparar los archivos con el comando “**git status**”



- Añadimos los archivos con **git add** . Comprobais que se añadido con **git status** y creamos el primer commit con **git commit -m “proyecto base hola mundo”**

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (master)
$ git add .
warning: LF will be replaced by CRLF in .idea/workspace.xml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/Main.java.
The file will have its original line endings in your working directory

andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (master)
$ git commit -m "proyecto base con hola mundo"
[master (root-commit) 2401e95] proyecto base con hola mundo
 8 files changed, 123 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .idea/misc.xml
 create mode 100644 .idea/modules.xml
 create mode 100644 .idea/vcs.xml
 create mode 100644 .idea/workspace.xml
 create mode 100644 README.md
 create mode 100644 ejercicioGit4.iml
 create mode 100644 src/Main.java
```

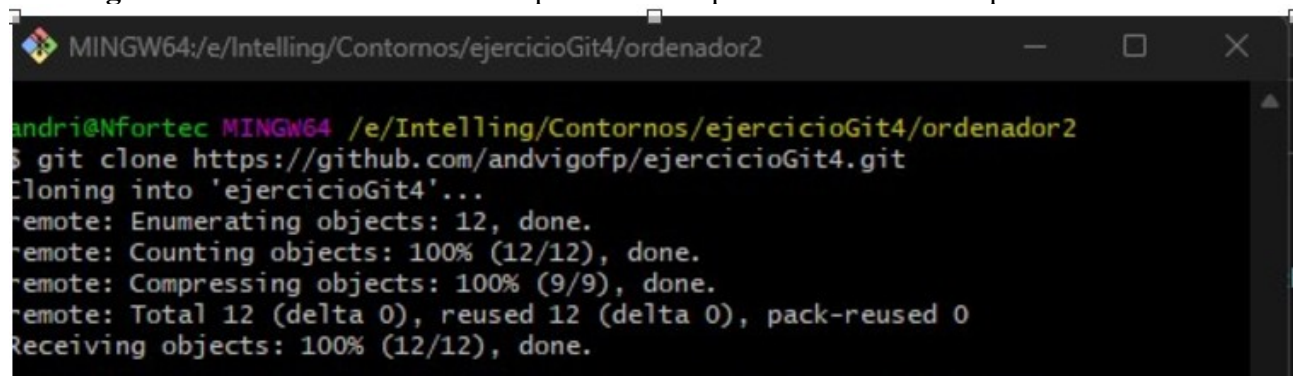
- Una vez creado el commit ahora nos toca subirlo al repo remoto
 - **git remote add origin** “la dirección de vuestro repositorio remoto”
 - **git push -u origin master** “solo la primera vez” . Para subir el proyecto a nuestro repositorio

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (master)
$ git remote add origin https://github.com/andvigofp/ejercicioGit4.git

andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (master)
$ git push -u origin master
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (12/12), 2.53 KiB | 1.26 MiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/andvigofp/ejercicioGit4.git
 * [new branch]      master -> master
```

6 - Clonamos el proyecto dentro de Ordenador2 y lo abrimos.

- Antes de clonar, tenemos que estar el directorio ordenador2
 - **git clone** “dirección de vuestro repo remoto” – para clonar vuestro repo en local



```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador2
$ git clone https://github.com/andvigofp/ejercicioGit4.git
Cloning into 'ejercicioGit4'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 12 (delta 0), reused 12 (delta 0), pack-reused 0
Receiving objects: 100% (12/12), done.
```

7 - Creamos una nueva funcionalidad, para lo cual creamos una rama "feat_persona" y la desarrollamos (ver mi log de commits). Una vez lista subimos la rama (git push feat_persona). Después, nos movemos a master para hacer un merge de "feat_persona" localmente. Al no haber nuevos commits en master, el merge debería ser automático y sin conflictos. Después del merge, hacemos un push de master con los nuevos cambios añadidos.

1. Primer paso crear una rama "feat_persona" y cambiar a rama feat_persona

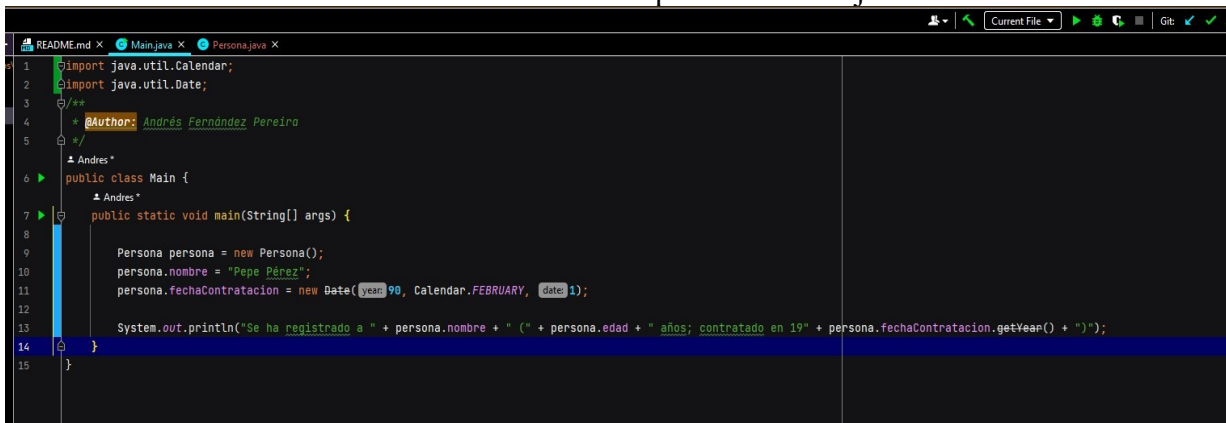
- git branch feat_persona – para crear una rama
- git checkout feat_persona – para cambiar de rama

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git branch feat_persona

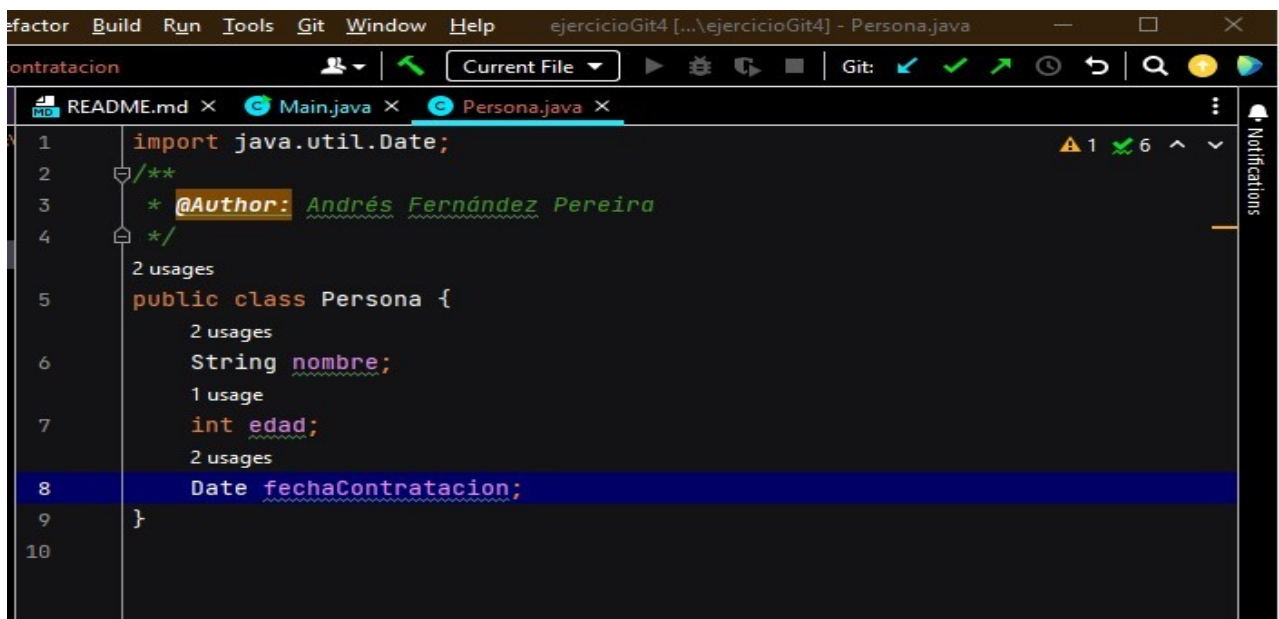
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git checkout feat_persona
Switched to branch 'feat_persona'
```

2. Segundo paso entrar en el proyecto en el ordenador/ejercicioGit4 editamos "main.java" y creamos el objeto persona "Persona.java"

- Modificamos el main "como está en commit probando un objeto Persona"



- Creamos el objeto persona "Persona.java"



- **git status** – Para comprobar estado de los archivos
- **git add .** - Para preparar los archivos existentes
- **git commit -m** “probando un objeto Persona” – Ahora lo tenemos preparado lo

```
a13andresfp@W10N-I8E02 MINGW64 /e/Dam/CONTORNO/IntelliJ/ejercicioGit4/ordenador2
/ejercicioGit4 (feat_persona)
$ git status
On branch feat_persona
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .idea/workspace.xml
        modified:   src/Main.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        src/Persona.java

no changes added to commit (use "git add" and/or "git commit -a")

a13andresfp@W10N-I8E02 MINGW64 /e/Dam/CONTORNO/IntelliJ/ejercicioGit4/ordenador2
/ejercicioGit4 (feat_persona)
$ git add .
warning: in the working copy of '.idea/workspace.xml', LF will be replaced by CR
LF the next time Git touches it

a13andresfp@W10N-I8E02 MINGW64 /e/Dam/CONTORNO/IntelliJ/ejercicioGit4/ordenador2
/ejercicioGit4 (feat_persona)
$ git commit -m "probando un objeto Persona"
[feat_persona f812e50] probando un objeto Persona
3 files changed, 46 insertions(+), 9 deletions(-)
create mode 100644 src/Persona.java
```

comitéamos

3. Ahora nos toca subir al remoto a la nueva rama **feat_persona**

- **git push --set-upstream origin feat_persona** – Porque no lo tenemos configurado automáticamente, utilizamos este comando.

```
a13andresfp@W10N-I8E02 MINGW64 /e/Dam/CONTORNO/IntelliJ/ejercicioGit4/ordenador2
/ejercicioGit4 (feat_persona)
$ git push
fatal: The current branch feat_persona has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feat_persona

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
```

4. Ahora nos toca cambiar de rama a **master** y mergear(fusinar ramas) de la rama **feat_persona** a **master**

- **git checkout master** – Cambiar de rama a master
- **git merge feat_persona** – Para fusionar la ramas

```
a13andresfp@W10N-I8E02 MINGW64 /e/Dam/CONTORNO/IntelliJ/ejercicioGit4/ordenador2
/ejercicioGit4 (feat_persona)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

a13andresfp@W10N-I8E02 MINGW64 /e/Dam/CONTORNO/IntelliJ/ejercicioGit4/ordenador2
/ejercicioGit4 (master)
$ git merge feat_persona
Updating c9fac88..f812e50
Fast-forward
 .idea/workspace.xml | 36 ++++++-----
 src/Main.java      | 10 ++++++
 src/Persona.java   | 9 ++++++
 3 files changed, 46 insertions(+), 9 deletions(-)
 create mode 100644 src/Persona.java
```

- Ultimo nos toca subir el commit que mergeamos de feat_persona
 - **git push** – Subir repo a remoto a la rama master

```
andri@Nfortec MINGW64 /e/IntelliJ/Contornos/ejercicioGit4/ordenador2/ejercicioG
it4 (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/andvigofp/ejercicioGit4.git
 d5c2d85..f8a86b8 master -> master
```

8- Volvemos a Ordenador1. Recordamos el estado de nuestro proyecto en este ordenador haciendo un **git status** y un **git log**. Si no nos indica que el master local está por detrás del remoto (origin/master), será necesario hacer un **git fetch**.

- **git status** – comprobar el estado del archivos
- **git log** - comprobar los commits
- **git fetch** – para actualizar los cambios del repositorio

```
a13andresfp@W10N-I8E02 MINGW64 /e/Dam/CONTORNO/IntelliJ/ejercicioGit4/ordenador1
/ejercicioGit4 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   .idea/workspace.xml

no changes added to commit (use "git add" and/or "git commit -a")

a13andresfp@W10N-I8E02 MINGW64 /e/Dam/CONTORNO/IntelliJ/ejercicioGit4/ordenador1
/ejercicioGit4 (master)
$ git log
commit c9fac88a807ee0f97c6d42ebce2188966fb0036f (HEAD -> master, origin/master)
Author: Andrés Fernández Pereira <animaccion96@gmail.com>
Date:   Mon Nov 28 20:48:56 2022 +0100

    proyecto base con hola mundo

a13andresfp@W10N-I8E02 MINGW64 /e/Dam/CONTORNO/IntelliJ/ejercicioGit4/ordenador1
/ejercicioGit4 (master)
$ git fetch
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 3), reused 7 (delta 3), pack-reused 0
Unpacking objects: 100% (7/7), 1.29 KiB | 7.00 KiB/s, done.
From https://github.com/andvigofp/ejercicioGit4
   c9fac88..f812e50 master -> origin/master
   * [new branch]      feat_persona -> origin/feat_persona
```

9- Para poder seguir trabajando desde Ordenador1, deberíamos hacer antes de nada un pull para traernos el nuevo commit que tiene master en remoto. Sin embargo, vamos a probar qué sucedería si nos olvidamos de ello.

- Este paso nos saltamos, lo normal había que hacer un git pull “para que descargue el proyecto, ramas y commit de remoto”.

10 - Desarrollamos una nueva funcionalidad "feat_empleado" en su correspondiente rama.

- `git branch feat_empleado` – creamos una nueva rama feat_empleado
- `git checkout feat_empleado` – nos cambiamos a la rama feat_empleado

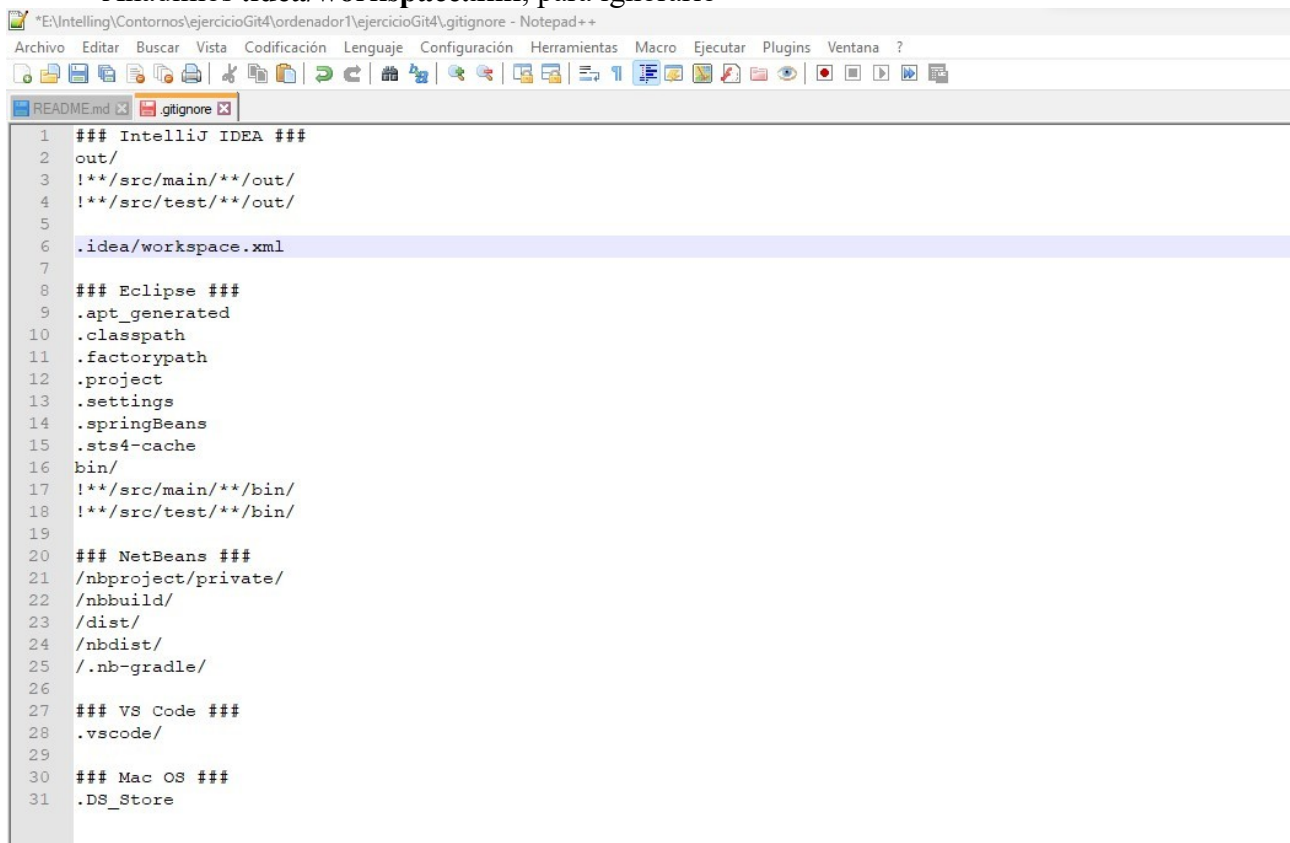
```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (master)
$ git branch feat_empleado

andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (master)
$ git checkout feat_empleado
Switched to branch 'feat_empleado'
M       .idea/workspace.xml
```

I. Durante el desarrollo, antes de hacer el primer commit en feat_empleado, detectamos que el fichero `.idea/workspace.xml` cambia automáticamente a menudo al abrir el proyecto en IntelliJ (es un archivo de metadatos de proyecto de IntelliJ).

- Editamos el archivo `.gitignore` del directorio `ordenador1/ejercicioGit4`:

- Añadimos `.idea/workspace.xml`, para ignorarlo



```
*E:\Intelling\Contornos\ejercicioGit4\ordenador1\ejercicioGit4\gitignore - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?

1  ### IntelliJ IDEA ###
2  out/
3  !**/src/main/**/out/
4  !**/src/test/**/out/
5
6  .idea/workspace.xml
7
8  ### Eclipse ###
9  .apt_generated
10 .classpath
11 .factorypath
12 .project
13 .settings
14 .springBeans
15 .sts4-cache
16 bin/
17 !**/src/main/**/bin/
18 !**/src/test/**/bin/
19
20 ### NetBeans ###
21 /nbproject/private/
22 /nbbuild/
23 /dist/
24 /nbdist/
25 /.nb-gradle/
26
27 ### VS Code ###
28 .vscode/
29
30 ### Mac OS ###
31 .DS_Store
```


II. Hacemos un primer commit para añadir .idea/workspace.xml al .gitignore. Además, como los ficheros de ese directorio ya se han añadido previamente al repositorio, tendremos que eliminarlos de él haciendo:

- Antes de crear el commit tenemos que borrar la cache archivo del **workspace.xml**.
 - **git rm --cached .idea/workspace.xml** – utilizamos este comando para borrar la cache archivo y su cache

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioG
it4 (feat_empleado)
$ git rm --cached .idea/workspace.xml
rm '.idea/workspace.xml'
```

- Ahora creamos el commit “añadido workspce.xml a gitignore”

- git status
- git add .
- git commit -m “añadido workspce.xml a gitignore”

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioG
it4 (feat_empleado)
$ git status
On branch feat_empleado
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    .idea/workspace.xml

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   .gitignore

andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioG
it4 (feat_empleado)
$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory

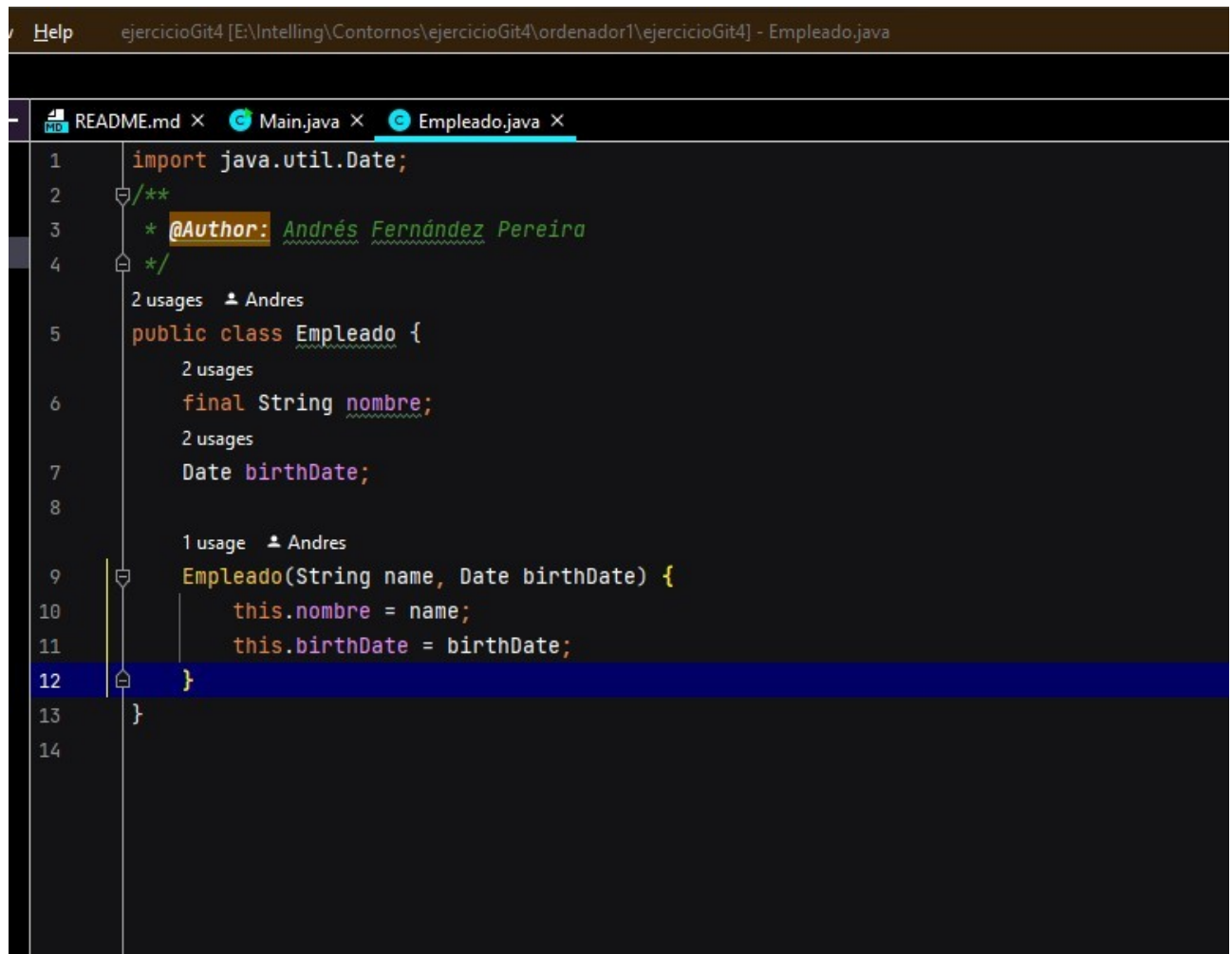
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioG
it4 (feat_empleado)
$ git status
On branch feat_empleado
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   .gitignore
    deleted:    .idea/workspace.xml

andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioG
it4 (feat_empleado)
$ git commit -m "añadido workspace.xml a gitignore"
[feat_empleado f769643] añadido workspace.xml a gitignore
2 files changed, 2 insertions(+), 33 deletions(-)
delete mode 100644 .idea/workspace.xml
```

III. Hacemos los otros 2 commits y pusheamos.

1. Primero creamos el objeto empleado **Empleado.java**, editamos el archivo **Main.java** y su commit “probando objeto empleado”

- Creamos el objeto empleado **Empleado.java**

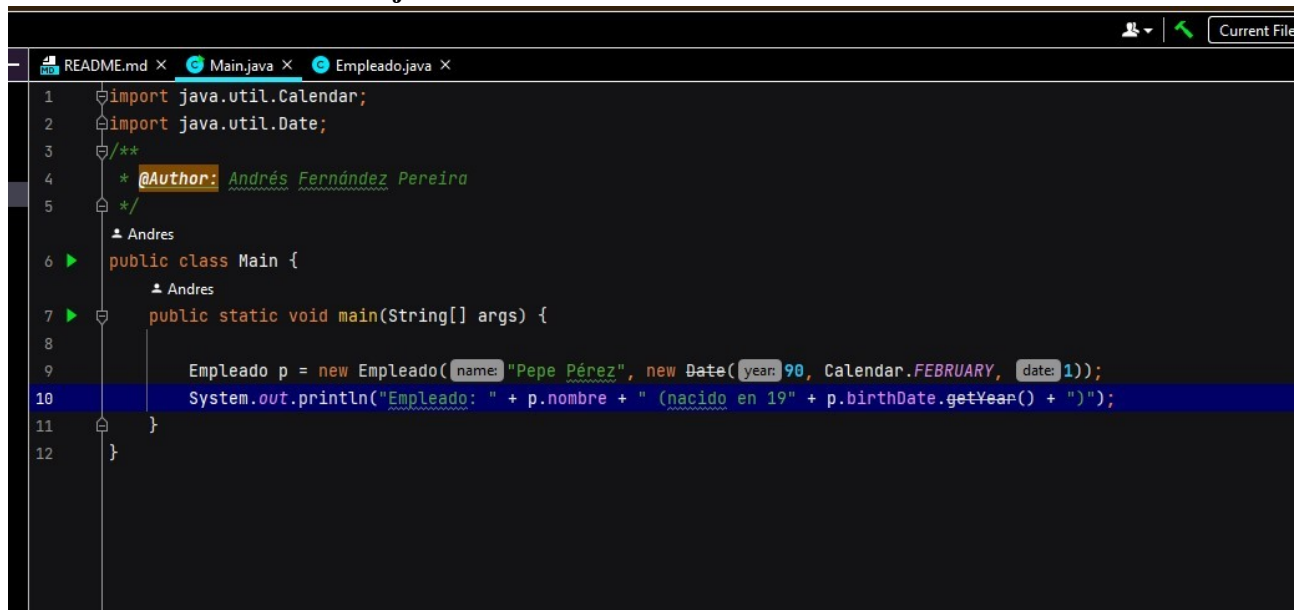


```
Help  ejercicioGit4 [E:\Intelling\Contornos\ejercicioGit4\ordenador1\ejercicioGit4] - Empleado.java

- README.md x Main.java x Empleado.java x

1  import java.util.Date;
2  /**
3   * @Author: Andrés Fernández Pereira
4   */
5  2 usages  Andres
   public class Empleado {
6       2 usages
       final String nombre;
7       2 usages
       Date birthDate;
8
9       1 usage  Andres
       Empleado(String name, Date birthDate) {
10           this.nombre = name;
11           this.birthDate = birthDate;
12       }
13   }
14
```

- Modificamos el archivo **Main.java**



```
1 import java.util.Calendar;
2 import java.util.Date;
3 /**
4  * @Author: Andrés Fernández Pereira
5  */
6 public class Main {
7     public static void main(String[] args) {
8
9         Empleado p = new Empleado("Pepe Pérez", new Date(year: 90, Calendar.FEBRUARY, date: 1));
10        System.out.println("Empleado: " + p.nombre + " (nacido en 19" + p.birthDate.getYear() + ")");
11    }
12 }
```

- **git status**
- **git add .**
- **git commit -m “probando objeto empleado”** – Añadimos el commit “probando objeto empleado”

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioG
it4 (feat_empleado)
$ git status
On branch feat_empleado
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   src/Main.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    src/Empleado.java

no changes added to commit (use "git add" and/or "git commit -a")

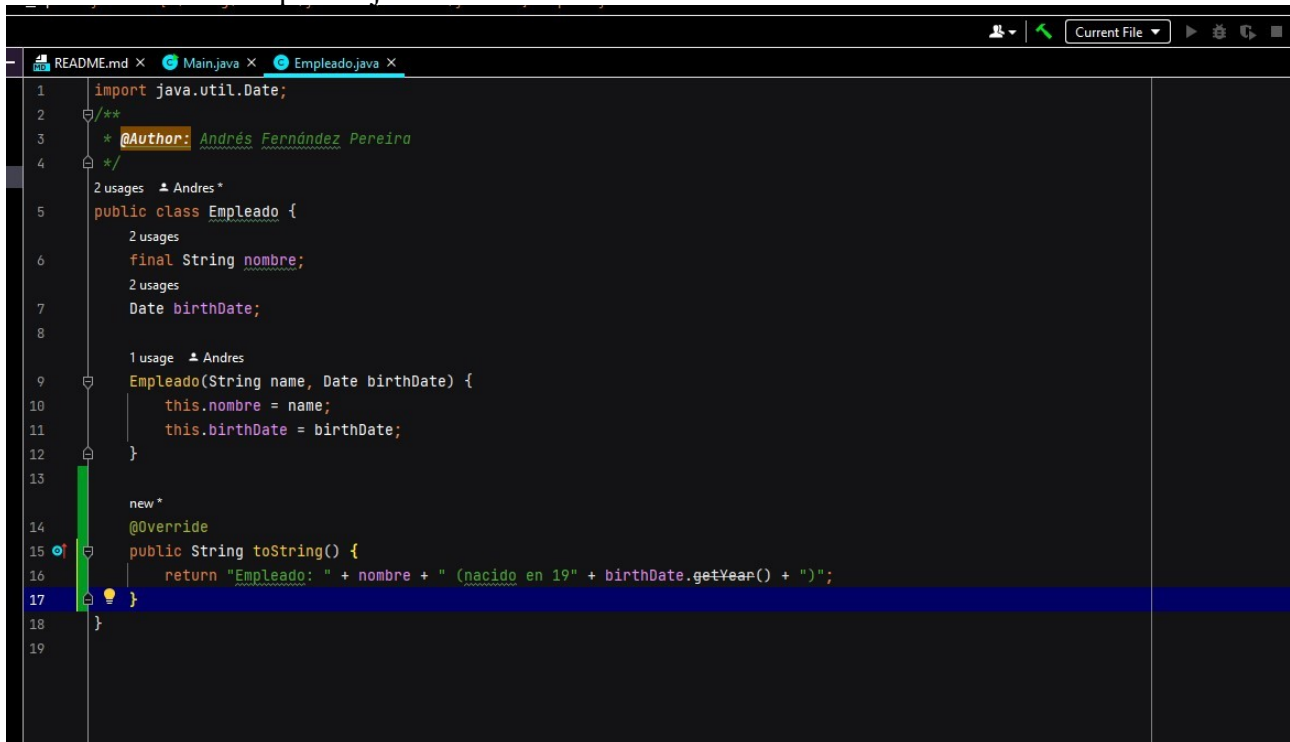
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (feat_empleado)
$ git add .
warning: LF will be replaced by CRLF in src/Main.java.
The file will have its original line endings in your working directory

andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (feat_empleado)
$ git status
On branch feat_empleado
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   src/Empleado.java
    modified:   src/Main.java

andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (feat_empleado)
$ git commit -m "probando objeto empleado"
[feat_empleado 0c8b5a9] probando objeto empleado
 2 files changed, 18 insertions(+), 1 deletion(-)
 create mode 100644 src/Empleado.java
```

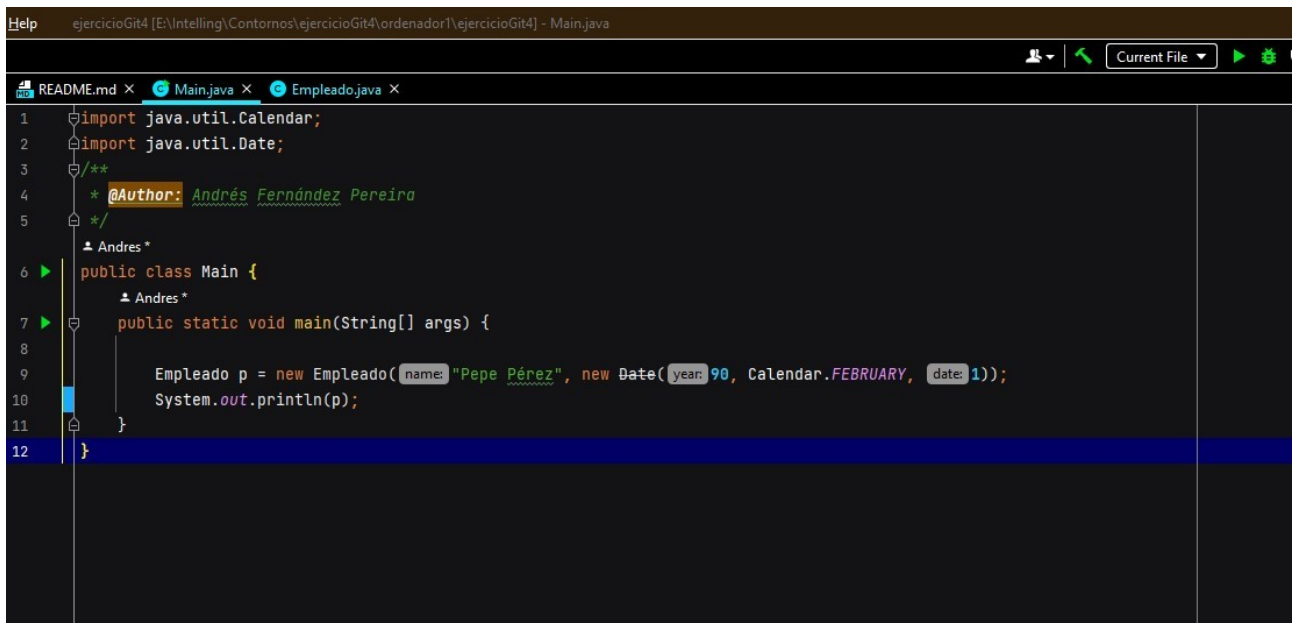
2. Modificamos el archivo **Empleado.java**, **Main.java** y último añadimos el **commit** “encapsulando toString de Empleado”.

- Modificamos el Empleado.java



```
1 import java.util.Date;
2 /**
3  * @Author: Andrés Fernández Pereira
4  */
5 public class Empleado {
6     final String nombre;
7     Date birthDate;
8
9     Empleado(String name, Date birthDate) {
10         this.nombre = name;
11         this.birthDate = birthDate;
12     }
13
14     @Override
15     public String toString() {
16         return "Empleado: " + nombre + " (nacido en 19" + birthDate.getYear() + ")";
17     }
18 }
19
```

- Modificamos el **Main.java**



```
1 import java.util.Calendar;
2 import java.util.Date;
3 /**
4  * @Author: Andrés Fernández Pereira
5  */
6 public class Main {
7     public static void main(String[] args) {
8
9         Empleado p = new Empleado("Pepe Pérez", new Date(year: 90, Calendar.FEBRUARY, date: 1));
10         System.out.println(p);
11     }
12 }
```


- **git status**
- **git add .**
- **git commit -m “encapsulando toString de Empleado”** – Añadimos el commit “encasulando toString de Empleado”

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (feat_empleado)
$ git status
On branch feat_empleado
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/Empleado.java
        modified:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")

andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (feat_empleado)
$ git add .
warning: LF will be replaced by CRLF in src/Main.java.
The file will have its original line endings in your working directory

andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (feat_empleado)
$ git commit -m "encapsulando toString de Empleado"
[feat_empleado 7621052] encapsulando toString de Empleado
2 files changed, 6 insertions(+), 1 deletion(-)
```

- Ahora nos toca subir los tres commits a la rama feat_empleado

- **git push --set-upstream origin feat_empleado**

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (feat_empleado)
$ git push
fatal: The current branch feat_empleado has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feat_empleado

andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (feat_empleado)
$ git push --set-upstream origin feat_empleado
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 12 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (14/14), 1.57 KiB | 802.00 KiB/s, done.
Total 14 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
remote:
remote: Create a pull request for 'feat_empleado' on GitHub by visiting:
remote:   https://github.com/andvigofp/ejercicioGit4/pull/new/feat_empleado
remote:
To https://github.com/andvigofp/ejercicioGit4.git
 * [new branch]      feat_empleado -> feat_empleado
Branch 'feat_empleado' set up to track remote branch 'feat_empleado' from 'origin'.
```

IV. **Vemos todas las ramas que tenemos.**

- **git branch -a** – para ver todas las ramas

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/ejercicioGit4 (feat_empleado)
$ git branch -a
* feat_empleado
  master
  remotes/origin/feat_empleado
  remotes/origin/feat_persona
  remotes/origin/master
```


feat_persona solo debería estar en remoto, ya que es una rama que no se ha usado en este "ordenador".

V. Hacemos un checkout a master para mergear. Al hacer el push detectamos que hay un conflicto. Hemos olvidado de descargarnos los cambios que habíamos hecho en el otro ordenador y que tenemos subidos a GitHub.

- Nos cambiamos de rama de feat_empleado a la rama master

- git checkout master

```
create mode 100644 src/Persona.java
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador1/ejercicioGit4 (
master)
$ git merge feat_empleado
CONFLICT (modify/delete): .idea/workspace.xml deleted in feat_empleado and modif
ied in HEAD. Version HEAD of .idea/workspace.xml left in tree.
Auto-merging src/Main.java
CONFLICT (content): Merge conflict in src/Main.java
Automatic merge failed; fix conflicts and then commit the result.
```

- Nos da un fallo al hacer un **push**, porque no hicimos desde el principio un **git pull** – para descargar el contenido del repositorio remoto a local en **ordenador1/ejercicioGit4**. Porque no hicimos el pull antes para descargar el contenido al ordenador1

- git push

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador1/EjercicioG
it4 (master)
$ git push
To https://github.com/andvigofp/ejercicioGit4.git
! [rejected]        master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/andvigofp/ejercicioGit4.g
it'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

VI. Podríamos hacer un pull, que implicaría un merge de la rama remota a la rama local, como ya hemos visto, pero para hacerlo bien, vamos a eliminar los nuevos commits de master (siguen en feat_empleado, así que no se pierden). Para ello, usamos un git reset --hard desde master, volviendo al último commit común al master remoto

- git reset --hard – Para eliminar los nuevos commit

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador1/ejercicioGit4 (
master)
$ git reset --hard
HEAD is now at bdcd29c proyecto base con hola mundo
```

VII. Ahora sí podemos hacer un pull de master sin conflicto.

- **git pull** – Ahora podemos descargar el contenido en remoto sin conflicto

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador1/ejercicioGit4 (
master)
$ git pull
Updating bdc29c..fd10c16
Fast-forward
 .idea/workspace.xml | 24 ++++++-----
 src/Main.java       | 10 ++++++
 src/Persona.java    | 9 ++++++
 3 files changed, 37 insertions(+), 6 deletions(-)
 create mode 100644 src/Persona.java
```

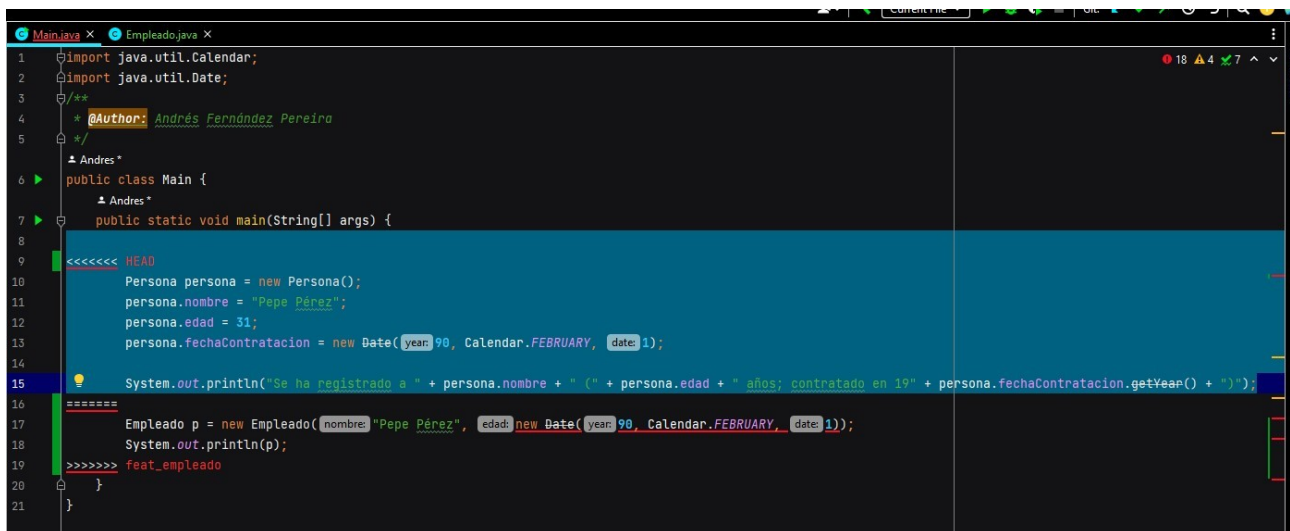
VIII. Hacemos merge de la nueva rama local feat_empleado. Ahora sí tendremos conflicto.

- **git merge feat_empleado**- Hacemos un merge y vemos que hay conflicto

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador1/ejercicioGit4 (
master)
$ git merge feat_empleado
CONFLICT (modify/delete): .idea/workspace.xml deleted in feat_empleado and modif
ied in HEAD. Version HEAD of .idea/workspace.xml left in tree.
Auto-merging src/Main.java
CONFLICT (content): Merge conflict in src/Main.java
Automatic merge failed; fix conflicts and then commit the result.
```

Solvamos los conflictos (en Main.java) y como realmente se está abordando la misma funcionalidad, lo llevamos más allá y dejamos una versión funcional con lo que queremos quedarnos de cada rama (ver en mi log).

- Tenemos que eliminar todo lo necesitamos en Main.java



```
1 import java.util.Calendar;
2 import java.util.Date;
3 /**
4  * @Author: Andrés Fernández Pereira
5  */
6 Andres*
7 public class Main {
8     Andres*
9     public static void main(String[] args) {
10
11         <<<<<< HEAD
12         Persona persona = new Persona();
13         persona.nombre = "Pepe Pérez";
14         persona.edad = 31;
15         persona.fechaContratacion = new Date(year 90, Calendar.FEBRUARY, date 1);
16
17         System.out.println("Se ha registrado a " + persona.nombre + " (" + persona.edad + " años; contratado en 19" + persona.fechaContratacion.getYear() + ")");
18
19         =====
20         Empleado p = new Empleado(nombre "Pepe Pérez", edad new Date(year 90, Calendar.FEBRUARY, date 1));
21         System.out.println(p);
22         >>>>>> feat_empleado
23     }
24 }
```

- Hacemos los cambios necesarios tenemos que dejar igual que el **commit**

```
Main.java x Empleado.java x
1 import java.util.Calendar;
2 import java.util.Date;
3 /**
4  * @Author: Andrés Fernández Pereira
5  */
6 public class Main {
7     Andres *
8     public static void main(String[] args) {
9
10         Empleado p = new Empleado( name: "Pepe Pérez", birthDate: 31, new Date( year: 90, Calendar.FEBRUARY, date: 1));
11         System.out.println(p);
12     }
13 }
14 }
```

- Este es el que tenemos por defecto el objeto Empleado.java

```
Empleado.java x
import java.util.Date;
/**
 * @Author: Andrés Fernández Pereira
 */
2 usages 1 related problem
class Empleado {
    2 usages
    final String nombre;
    2 usages
    Date birthDate;

    1 usage 1 related problem
    Empleado(String name, Date birthDate) {
        this.nombre = name;
        this.birthDate = birthDate;
    }

    @Override
    public String toString() {
        return "Empleado: " + nombre + " (nacido en 19" + birthDate.getYear() + ")";
    }
}
```

```
va x Empleado.java x
* @Author: Andrés Fernández Pereira
*/
2 usages
class Empleado {
    2 usages
    String nombre;
    2 usages
    int edad;
    2 usages
    Date fechaContratacion;

    1 usage
    public Empleado(String nombre, int edad, Date fechaContratacion) {
        this.nombre = nombre;
        this.edad = edad;
        this.fechaContratacion = fechaContratacion;
    }

    @Override
    public String toString() {
        return "Se ha registrado a " + nombre + " (" + edad + " años; contratado en 19" + fechaContratacion.getYear() + ")";
    }
}
```

- Al comprobar con `git status` comprobamos que nos está añadiendo **.idea/workspace.xml**. Este conflicto pasa porque tendríamos que haber ignorado desde el principio y eliminar la cache **workspace.xml** y el conflicto viene desde el primer commit.

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador1/ejercicioGit4 (
master|MERGING)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  modified:   .gitignore
  new file:   src/Empleado.java

Unmerged paths:
  (use "git add/rm <file>..." as appropriate to mark resolution)
    deleted by them: .idea/workspace.xml
    both modified:   src/Main.java

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   src/Empleado.java
```

- Eliminamos la caché con `git rm --cached .idea/workspace.xml`

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador1/ejercicioGit4 (
master|MERGING)
$ git rm --cached .idea/workspace.xml
rm '.idea/workspace.xml'
```

IX. Terminamos el merge con add y commit, y hacemos un push.

- `git status`
- `git add .`
- `git commit -m "Merge remote-tracking branch 'origin/feat_empleado'"`
- `git push` – una vez echo todos los pasos anteriores ya podemos subir el contenido a la rama master

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador1/ejercicioGit4 (master|MERGING)
$ git add .

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador1/ejercicioGit4 (master|MERGING)
$ git commit -m "Merge remote-tracking branch 'origin/feat_empleado'"
[master 5dac350] Merge remote-tracking branch 'origin/feat_empleado'

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador1/ejercicioGit4 (master)
$ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 711 bytes | 711.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/andvigofp/ejercicioGit4.git
fd10c16..5dac350 master -> master
```

11 -Volvemos a Ordenador2 y comprobamos si tenemos todo al día con `git status` y `git log`. Si no nos indica que el master local está por detrás del remoto, será necesario hacer un `git fetch`, que ahora nos indicará que debemos hacer un `pull`.

- Nos vamos al directorio `ordenador2/ejercicioGit4`, hacemos lo que nos indica

- `git status`
- `git log`
- `git fetch`

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git log
commit fd10c1612bb1a77da3776f658cdfbb5bb1d29bf3 (HEAD -> master, origin/master, origin/feat_persona, origin/HEAD, feat_persona)
Author: andvigo <animacion96@gmail.com>
Date: Thu Dec 1 19:45:26 2022 +0100

    probando un objeto Persona

commit bdc29c5f396e985dfc0dbed9637efff1040b96f
Author: andvigo <animacion96@gmail.com>
Date: Thu Dec 1 19:36:02 2022 +0100

    proyecto base con hola mundo

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git fetch
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 19 (delta 9), reused 18 (delta 8), pack-reused 0
Unpacking objects: 100% (19/19), 2.06 KiB | 1024 bytes/s, done.
From https://github.com/andvigofp/ejercicioGit4
fd10c16..5dac350 master -> origin/master
* [new branch] feat_empleado -> origin/feat_empleado
```


- Por último un **git pull** – para descargar el contenido a remoto.

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git pull
Updating fd10c16..5dac350
Fast-forward
 .gitignore          | 2 ++
 .idea/workspace.xml | 47 -----
 src/Empleado.java   | 22 +++++
 src/Main.java       | 8 +++
 src/Persona.java    | 9 -----
5 files changed, 27 insertions(+), 61 deletions(-)
delete mode 100644 .idea/workspace.xml
create mode 100644 src/Empleado.java
delete mode 100644 src/Persona.java
```

- Si los diera algún fallo como me ocurrió, en alguna de las pruebas que hice, los traigo la solución.
 - **git pull** – Como veis me da un fallo con un conflicto con una de las ramas

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador2/ejercicioGit4b (master)
$ git pull
error: Your local changes to the following files would be overwritten by merge:
 .idea/workspace.xml
Please commit your changes or stash them before you merge.
Aborting
Updating 709af99..6a840e9
```

- Hací un **git reset --hard** – Para que vuelva al último commit

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador2/ejercicioGit4b (master)
$ git reset --hard
HEAD is now at 709af99 probando un objeto Persona
```

- Ahora ya podéis hacer **git pull** – Ya se arreglo el problema y los deja descargar el contenido con sus ramas correspondientes y commits

```
andri@Nfortec MINGW64 /e/Intelling/Contornos/ejercicioGit4/ordenador2/ejercicioGit4b (master)
$ git pull
Updating 709af99..6a840e9
Fast-forward
 .gitignore          | 2 ++
 .idea/workspace.xml | 19 +++++
 src/Empleado.java   | 21 +++++
 src/Main.java       | 8 +++
4 files changed, 38 insertions(+), 12 deletions(-)
create mode 100644 src/Empleado.java
```

12 -Ahora vamos a simular cómo se desarrollaría en paralelo en varias ramas (distintas personas). No crearemos carpetas que simulen ordenadores pero cada rama representará el trabajo independiente de cada componente del equipo de trabajo.

- I. Creamos dos ramas desde master, que desarrollaremos sin hacer merge a master (simulando otro desarrollador trabajando en paralelo):
- Creamos a los ramas correspondiente **feat_inmuebles_filelds**(Aquí me equivoque y lo edito la rama en remoto) y último creamos a la otra rama **bug_edad_negativa**

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git branch feat_inmutables
```

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git branch bug_edad_negativa
```

- A) "feat_inmutable_fields" para evitar que se pueda cambiar el nombre y la fecha de contratación de un empleado (asumimos que las especificaciones del proyecto lo impiden, sea o no razonable).

- Nos cambiamos de rama con "git checkout feat_inmutables"

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git checkout feat_inmutables
Switched to branch 'feat_inmutables'
```

- Editamos el archivo **Main.java** y lo ponemos como en el **commit** "diseñamos las pruebas de esta funcionalidad"

```
Empleado p = new Empleado( nombre: "Pepe Pérez", edad: 31, new Date( year: 90, Calendar.FEBRUARY, date: 1));
System.out.println(p);

p.nombre = "Manolo"; // Cambio de nombre. No se debe permitir
p.fechaContratacion = new Date( year: 99, Calendar.FEBRUARY, date: 1); // Cambio de fecha. No se debe permitir
```

- Como siempre comprobamos los archivos que va añadir, añadimos el archivo modificado **Man.java** y último el **commit**

- **git status**
- **git add .**
- **git commit -m "diseñamos las pruebas de esta funcionalidad"**

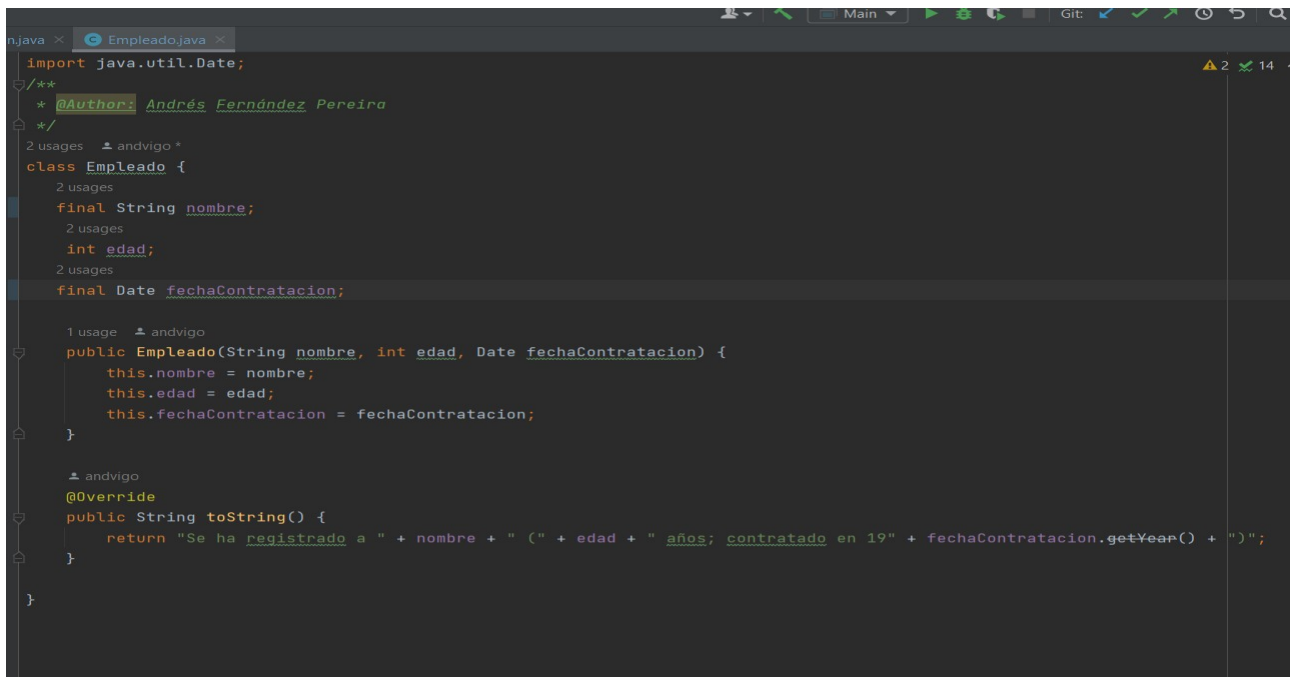
```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_inmutables)
$ git status
On branch feat_inmutables
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_inmutables)
$ git add .

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_inmutables)
$ git commit -m "diseñamos las pruebas de esta funcionalidad"
[feat_inmutables d2813af] diseñamos las pruebas de esta funcionalidad
1 file changed, 2 insertions(+)
```

- Editamos el archivo **Main.java** y **Empleado.java**, lo dejamos como el commit “**ahora hay dos campos inmutables**”

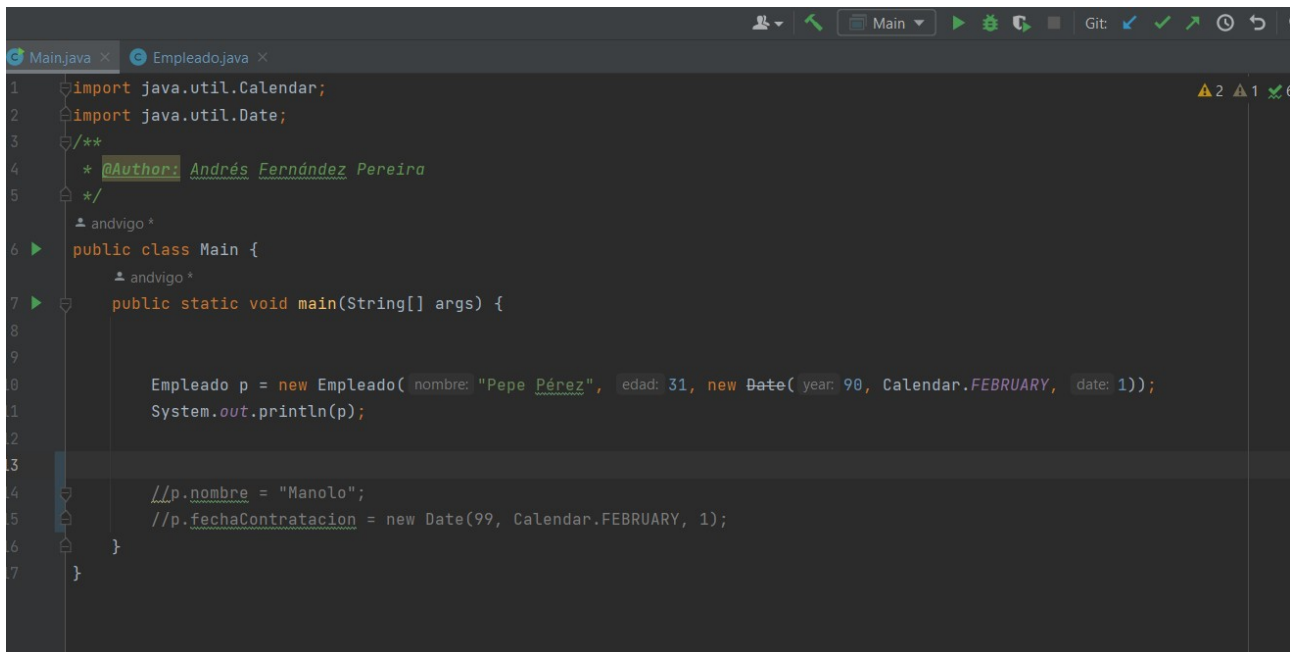


```
import java.util.Date;

/**
 * @Author: Andrés Fernández Pereira
 */
class Empleado {
    final String nombre;
    final int edad;
    final Date fechaContratacion;

    public Empleado(String nombre, int edad, Date fechaContratacion) {
        this.nombre = nombre;
        this.edad = edad;
        this.fechaContratacion = fechaContratacion;
    }

    @Override
    public String toString() {
        return "Se ha registrado a " + nombre + " (" + edad + " años; contratado en 19" + fechaContratacion.getYear() + ")";
    }
}
```



```
import java.util.Calendar;
import java.util.Date;

/**
 * @Author: Andrés Fernández Pereira
 */
public class Main {
    public static void main(String[] args) {
        Empleado p = new Empleado( nombre: "Pepe Pérez", edad: 31, new Date( year: 90, Calendar.FEBRUARY, date: 1));
        System.out.println(p);

        //p.nombre = "Manolo";
        //p.fechaContratacion = new Date(99, Calendar.FEBRUARY, 1);
    }
}
```

- Una vez ya creado el commit ahora nos toca subir el contenido a la rama correspondiente.

- **git status**
- **git add .**
- **git commit -m “ahora hay dos campos inmutables”**

```

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_inmutables)
$ git status
On branch feat_inmutables
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/Empleado.java
        modified:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_inmutables)
$ git add .

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_inmutables)
$ git commit -m "ahora hay dos campos inmutables"
[feat_inmutables f53dc2a] ahora hay dos campos inmutables
2 files changed, 5 insertions(+), 4 deletions(-)

```

- Subimos el contenido a la rama feat_inmutables
 - **git push --set-upstream origin feat_inmutables**

```

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git push --set-upstream origin feat_inmutables

```

B. "bug_edad_negativa", para garantizar que la edad de un empleado no pueda ser negativa.

- Nos cambiamos a la rama bug_edad_negativa **"git checkout bug_edad_negativa"**

```

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git checkout bug_edad_negativa
Switched to branch 'bug_edad_negativa'

```

- Modificamos el archivo Main.java, lo dejamos como el commit **"diseñamos la prueba que muestra el error"**

```

p.edad = -31; // Ahora acepta edades negativas

Empleado p2 = new Empleado( nombre: "Pepe Pérez", edad: -31, new Date( year: 90, Calendar.FEBRUARY, date: 1));
System.out.println(p);

```

- Hacemos como siempre, comprobamos el estado, añadimos y comiteamos.
 - **git status**
 - **git add .**
 - **git commit -m "diseñamos la prueba que muestra error"**

```

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_edad_negativa)
$ git status
On branch bug_edad_negativa
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_edad_negativa)
$ git add .

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_edad_negativa)
$ git commit -m "diseñamos la prueba que muestra el error"
[bug_edad_negativa 7adf4c5] diseñamos la prueba que muestra el error
1 file changed, 5 insertions(+)

```

- Modificamos el Main.java/Empleado.Java, lo dejamos como en el commit “**information hiding: edad es solo modificable a través de un setter**”

```

Main.java x Empleado.java x
4 usages andvigo *
class Empleado {
2 usages
String nombre;
3 usages
private int edad;
2 usages
Date fechaContratacion;

2 usages andvigo
public Empleado(String nombre, int edad, Date fechaContratacion) {
    this.nombre = nombre;
    this.edad = edad;
    this.fechaContratacion = fechaContratacion;
}

new *
public void setEdad(int edad) {
    if (edad < 0) throw new RuntimeException("Un empleado no puede tener edad negativa");
    else this.edad = edad;
}

andvigo
@Override
public String toString() {
    return "Se ha registrado a " + nombre + " (" + edad + " años; contratado en 19" + fechaContratacion.getYear() + ")";
}
}

```

```

Main.java x Empleado.java x
import java.util.Calendar;
import java.util.Date;
/**
 * @Author: Andrés Fernández Pereira
 */
andvigo *
public class Main {
    andvigo *
    public static void main(String[] args) {

        Empleado p = new Empleado( nombre: "Pepe Pérez", edad: 31, new Date( year: 90, Calendar.FEBRUARY, date: 1));
        System.out.println(p);

        //p.edad = -31; // Arreglado: error de compilación. edad no es accesible

        Empleado p2 = new Empleado( nombre: "Pepe Pérez", edad: -31, new Date( year: 90, Calendar.FEBRUARY, date: 1));
        System.out.println(p);

    }
}

```


- git status
- git add .
- git commit -m “información hidding: edad es solo modificado a través de un setter”

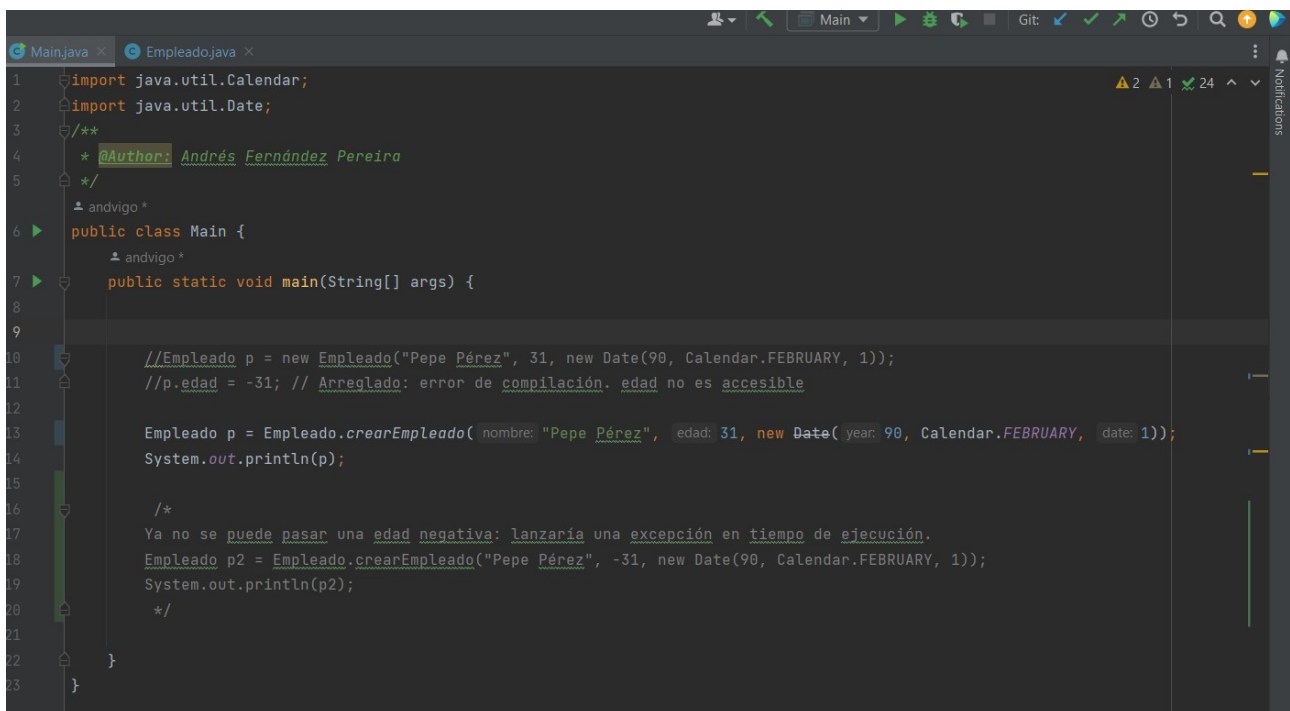
```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_edad_negativa)
$ git status
On branch bug_edad_negativa
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/Empleado.java
        modified:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_edad_negativa)
$ git add .

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_edad_negativa)
$ git commit -m "information hidding: edad es solo modificable a través de un setter"
[bug_edad_negativa e15a1b5] information hidding: edad es solo modificable a través de un setter
2 files changed, 6 insertions(+), 3 deletions(-)
```

- Ahora nos falta el último commit “**resuelto**”, tendremos que modificar el **Main.java** y **Empleado.java**



```
1 import java.util.Calendar;
2 import java.util.Date;
3 /**
4  * @Author: Andrés Fernández Pereira
5  */
6 public class Main {
7     public static void main(String[] args) {
8
9
10        //Empleado p = new Empleado("Pepe Pérez", 31, new Date(90, Calendar.FEBRUARY, 1));
11        //p.edad = -31; // Arreglado: error de compilación. edad no es accesible
12
13        Empleado p = Empleado.crearEmpleado( nombre: "Pepe Pérez", edad: 31, new Date( year: 90, Calendar.FEBRUARY, date: 1));
14        System.out.println(p);
15
16        /*
17        Ya no se puede pasar una edad negativa: lanzaría una excepción en tiempo de ejecución.
18        Empleado p2 = Empleado.crearEmpleado("Pepe Pérez", -31, new Date(90, Calendar.FEBRUARY, 1));
19        System.out.println(p2);
20        */
21    }
22 }
23 }
```

```
1  import java.util.Date;
2  /**
3   * @Author: Andrés Fernández Pereira
4   */
5  class Empleado {
6      String nombre;
7      private int edad;
8      Date fechaContratacion;
9
10     private Empleado(String nombre, int edad, Date fechaContratacion) {
11         this.nombre = nombre;
12         this.edad = edad;
13         this.fechaContratacion = fechaContratacion;
14     }
15     public void setEdad(int edad) {
16         if (edad < 0) throw new RuntimeException("Un empleado no puede tener edad negativa");
17         else this.edad = edad;
18     }
19     @Override
20     public String toString() {
21         return "Se ha registrado a " + nombre + " (" + edad + " años; contratado en 19" + fechaContratacion.getYear() + ")";
22     }
23
24     /**
25      * Método factoría que permite comprobar si la edad es válida antes de llamar al constructor.
26      * El constructor es ahora privado, impidiendo llamarlo directamente desde otra clase.
27      */
28     public static Empleado crearEmpleado(String nombre, int edad, Date fechaContratacion) {
29         if (edad < 0) throw new RuntimeException("Un empleado no puede tener edad negativa");
30         else return new Empleado(nombre, edad, fechaContratacion);
31     }
32 }
```

- Último como siempre comprobamos el estado del contenido, añadimos y comiteamos.

- git stat us
- git add .
- git commit -m "resuelto"

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
bug_edad_negativa)
$ git status
```

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
bug_edad_negativa)
$ git add .
```

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
bug_edad_negativa)
$ git commit -m "resuelto"
```

- Ahora nos toca subir el contenido a la rama **bug_edad_negativa**
 - **git push --set-upstream origin bug_edad_negativa**

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_edad_negativa)
$ git push --set-upstream origin bug_edad_negativa
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 12 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (14/14), 1.95 KiB | 1.95 MiB/s, done.
Total 14 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), completed with 1 local object.
remote:
remote: Create a pull request for 'bug_edad_negativa' on GitHub by visiting:
remote:   https://github.com/andvigofp/ejercicioGit4/pull/new/bug_edad_negativa
remote:
To https://github.com/andvigofp/ejercicioGit4.git
 * [new branch]      bug_edad_negativa -> bug_edad_negativa
branch 'bug_edad_negativa' set up to track 'origin/bug_edad_negativa'.
```

II. Una vez desarrolladas, hacemos el merge de ambas ramas a master solventando conflictos.

- Volvemos a la rama master
 - **git checkout master**

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_edad_negativa)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

- Ahora nos toca mergear para juntar la ramas a master

- **git merge feat_inmutables**

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git merge feat_inmutables
Updating 5dac350..f53dc2a
Fast-forward
 src/Empleado.java | 4 ++--
 src/Main.java      | 3 +++
 2 files changed, 5 insertions(+), 2 deletions(-)
```

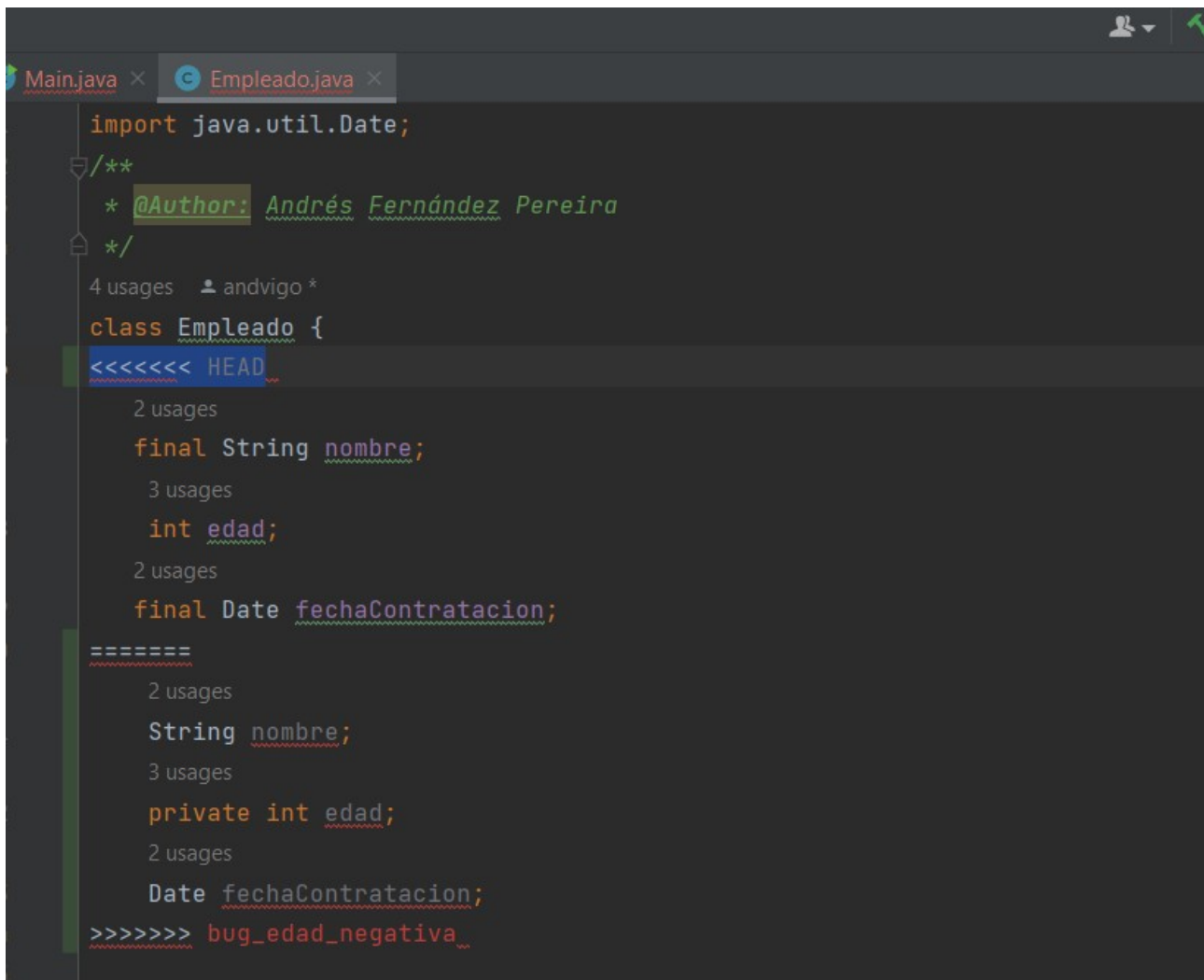
- **git push** – Ahora lo subimos a remoto a la rama master

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/andvigofp/ejercicioGit4.git
 5dac350..f53dc2a master -> master
```

- **git merge bug_edad_negativa** – Comprobamos que hay un conflicto al intentar juntar la rama bug_edad_negativa con master, tenemos que resolver los conflictos

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git merge bug_edad_negativa
Auto-merging src/Empleado.java
CONFLICT (content): Merge conflict in src/Empleado.java
Auto-merging src/Main.java
CONFLICT (content): Merge conflict in src/Main.java
Automatic merge failed; fix conflicts and then commit the result.
```

- Hay que arreglar el conflicto en Main.java y Empleado.java y dejarlo como en commit “**Merge branch 'bug_edad_negativa'**”



```
import java.util.Date;

/**
 * @Author: Andrés Fernández Pereira
 */
4 usages andvigo *
class Empleado {
<<<<<< HEAD
2 usages
final String nombre;
3 usages
int edad;
2 usages
final Date fechaContratacion;

=====
2 usages
String nombre;
3 usages
private int edad;
2 usages
Date fechaContratacion;
>>>>>> bug_edad_negativa
```

```
Main.java x Empleado.java x
import java.util.Date;

/**
 * @Author: Andrés Fernández Pereira
 */
4 usages andvigo *
class Empleado {
<<<<<<< HEAD
2 usages
final String nombre;
3 usages
int edad;
2 usages
final Date fechaContratacion;

=====
2 usages
String nombre;
3 usages
private int edad;
2 usages
Date fechaContratacion;
>>>>>>> bug_edad_negativa
```

```
14 src/Main.java
@@ -7,11 +7,23 @@ public class Main {
7 7 public static void main(String[] args) {
8 8
9 9
10 - Empleado p = new Empleado("Pepe Pérez", 31, new Date(90, Calendar.FEBRUARY, 1));
10 + //Empleado p = new Empleado("Pepe Pérez", 31, new Date(90, Calendar.FEBRUARY, 1));
11 + //p.edad = -31; // Arreglado: error de compilación. edad no es accesible
12 +
13 + Empleado p = Empleado.crearEmpleado("Pepe Pérez", 31, new Date(90, Calendar.FEBRUARY, 1));
11 14 System.out.println(p);
12 15
13 16
14 17 //p.nombre = "Manolo";
15 18 //p.fechaContratacion = new Date(99, Calendar.FEBRUARY, 1);
19 +
20 + /*
21 + Ya no se puede pasar una edad negativa: lanzaría una excepción en tiempo de ejecución.
22 + Empleado p2 = Empleado.crearEmpleado("Pepe Pérez", -31, new Date(90, Calendar.FEBRUARY, 1));
23 + System.out.println(p2);
24 + */
25 +
26 +
27 +
```



```

18  src/Empleado.java
3   3   * @Author: Andrés Fernández Pereira
4   4   */
5   5   class Empleado {
6   6   +
6   7   final String nombre;
7   8   int edad;
8   9   final Date fechaContratacion;
9   10
10  10  - public Empleado(String nombre, int edad, Date fechaContratacion) {
11  11  +
12  12  + private Empleado(String nombre, int edad, Date fechaContratacion) {
13  13      this.nombre = nombre;
14  14      this.edad = edad;
15  15      this.fechaContratacion = fechaContratacion;
16  16  }
17  17
18  18  -
19  19  + public void setEdad(int edad) {
20  20  +     if (edad < 0) throw new RuntimeException("Un empleado no puede tener edad negativa");
21  21  +     else this.edad = edad;
22  22  + }
23  23
24  24  @Override
25  25  public String toString() {
26  26      return "Se ha registrado a " + nombre + " (" + edad + " años; contratado en " + fechaContratacion.getYear() + ")";
27  27  }
28  28
29  29  +
30  30  + /**
31  31  +     * Método factoría que permite comprobar si la edad es válida antes de llamar al constructor.
32  32  +     * El constructor es ahora privado, impidiendo llamarlo directamente desde otra clase.
33  33  +     */
34  34  + public static Empleado crearEmpleado(String nombre, int edad, Date fechaContratacion) {
35  35  +     if (edad < 0) throw new RuntimeException("Un empleado no puede tener edad negativa");
36  36  +     else return new Empleado(nombre, edad, fechaContratacion);
37  37  + }
38  38
39  39  +
40  40  +

```

- Una vez echo los cambios nos queda añadir y comitear, último subirlo el contenido a la rama master

- **git status**
- **git add .**
- **git commit -m "Mege branch 'bug_edad_negativa'"**
- **git push**

```

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master|MERGING)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   src/Empleado.java
    both modified:   src/Main.java

```

```

no changes added to commit (use "git add" and/or "git commit -a")

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master|MERGING)
$ git add .

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master|MERGING)
$ git commit -m "Merge branch 'bug_edad_negativa'"
[master f691c9c] Merge branch 'bug_edad_negativa'

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 567 bytes | 567.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/andvigofp/ejercicioGit4.git
f53dc2a..f691c9c master -> master

```

13- Realizamos otras mejoras en la rama feat_pojo (Esta parte se olvido el profesor, hay que hacerlo antes de la última rama)

- Creamos la rama y después entramos en la rama feat_pojo
 - `git branch feat_pojo` o `git checkout -b feat_pojo`

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
master)
$ git branch feat_pojo
```

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
master)
$ git checkout feat_pojo
Switched to branch 'feat_pojo'
M       src/Main.java
```

- Modificamos **Main.java**, lo dejamos como el commit “mostrando el problema”

```
src/Main.java

@@ -23,6 +23,8 @@ public static void main(String[] args) {
    System.out.println(p2);
    /*
+   System.out.println("Si quiero mostrar solo el nombre: " + p.nombre);
+   System.out.println("Si quiero mostrar solo la edad no puedo: " + p.edad); // necesito un getter
    */
}
```

- `git status`
- `git add .`
- `git commit -m “mostrando el problema”`

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
feat_pojo)
$ git status
On branch feat_pojo
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   src/Main.java

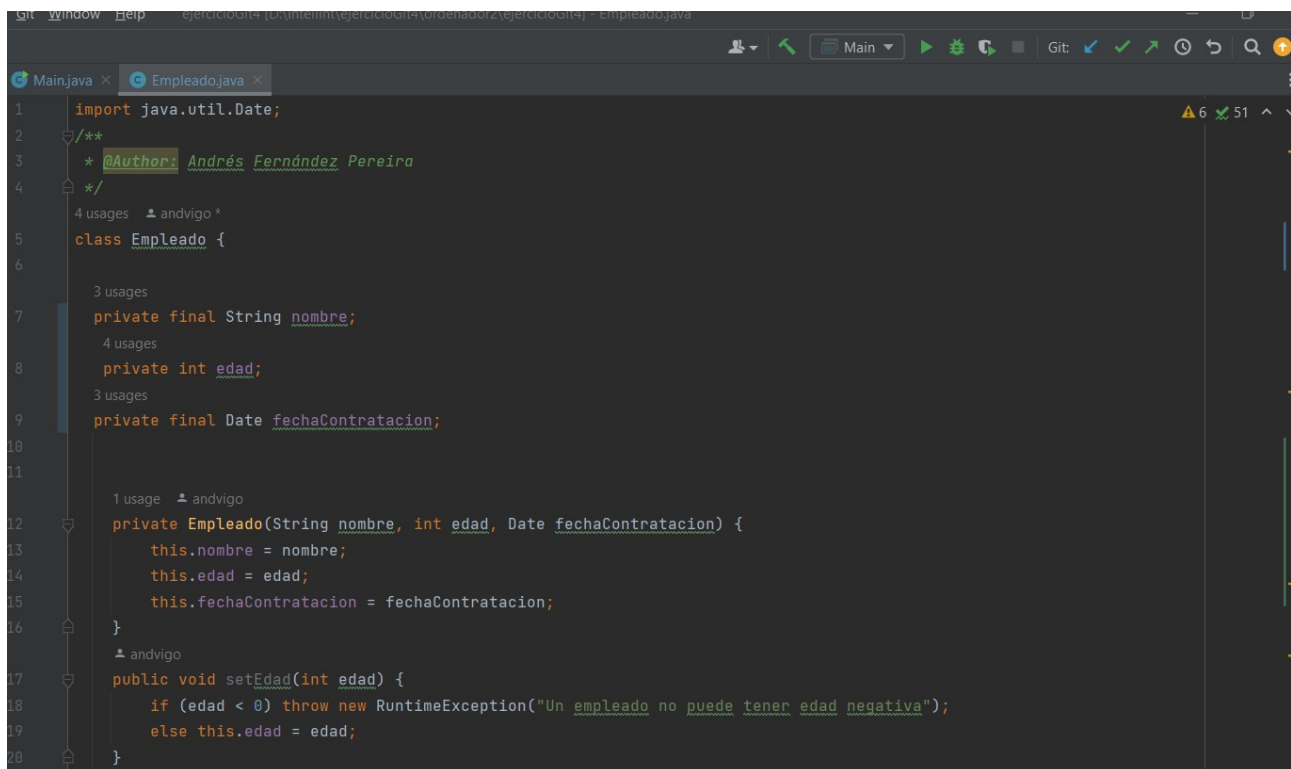
no changes added to commit (use "git add" and/or "git commit -a")

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
feat_pojo)
$ git add .

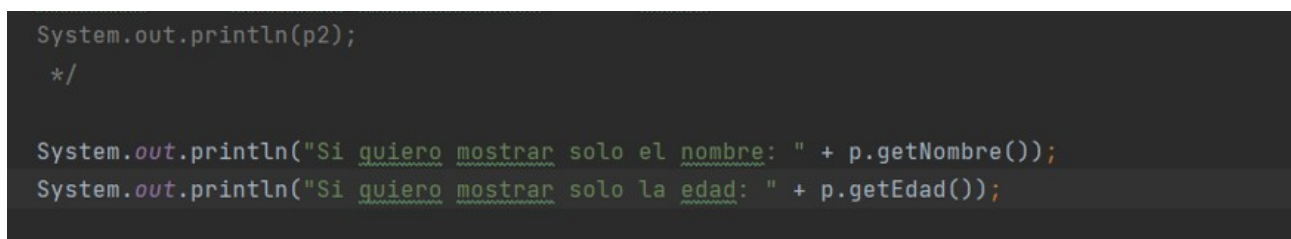
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
feat_pojo)
$ git status
On branch feat_pojo
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   src/Main.java

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
feat_pojo)
$ git commit -m "mostrando el problema"
[feat_pojo a71c595] mostrando el problema
1 file changed, 2 insertions(+)
```

- Tenemos que modificar el **Main.java** y **Empleado.java**, como siempre dejarlo como el commit correspondiente “**los campos son privados y se accede mediante getters**”



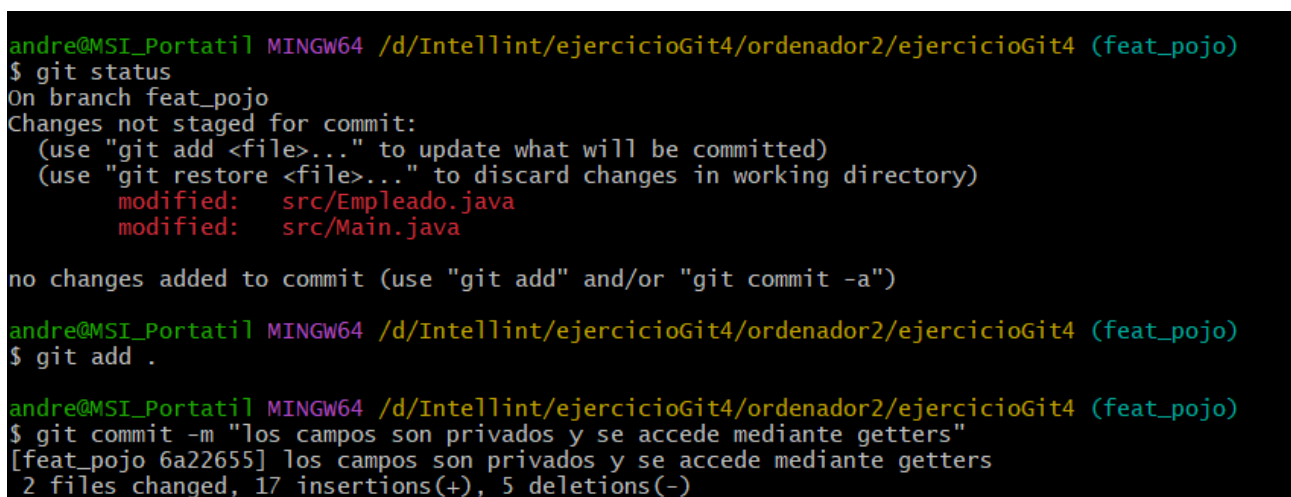
```
1 import java.util.Date;
2 /**
3  * @Author: Andrés Fernández Pereira
4  */
5 class Empleado {
6
7     private final String nombre;
8     private int edad;
9     private final Date fechaContratacion;
10
11     private Empleado(String nombre, int edad, Date fechaContratacion) {
12         this.nombre = nombre;
13         this.edad = edad;
14         this.fechaContratacion = fechaContratacion;
15     }
16
17     public void setEdad(int edad) {
18         if (edad < 0) throw new RuntimeException("Un empleado no puede tener edad negativa");
19         else this.edad = edad;
20     }
21 }
```



```
System.out.println(p2);
*/

System.out.println("Si quiero mostrar solo el nombre: " + p.getNombre());
System.out.println("Si quiero mostrar solo la edad: " + p.getEdad());
```

- **git status**
- **git add .**
- **git commit -m “los campos son privados y se accede mediante getters”**



```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_pojo)
$ git status
On branch feat_pojo
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   src/Empleado.java
    modified:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_pojo)
$ git add .

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_pojo)
$ git commit -m "los campos son privados y se accede mediante getters"
[feat_pojo 6a22655] los campos son privados y se accede mediante getters
2 files changed, 17 insertions(+), 5 deletions(-)
```

- Lo subimos a la rama con el comando **git push --set-upstream origin feat_pojo**

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_pojo)
$ git push --set-upstream origin feat_pojo
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 1.02 KiB | 1.02 MiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 3 local objects.
remote:
remote: Create a pull request for 'feat_pojo' on GitHub by visiting:
remote:   https://github.com/andvigofp/ejercicioGit4/pull/new/feat_pojo
remote:
To https://github.com/andvigofp/ejercicioGit4.git
 * [new branch]      feat_pojo -> feat_pojo
branch 'feat_pojo' set up to track 'origin/feat_pojo'.
```

- Ultimo regresamos a la rama master, lo mergeamos, para juntar la rama con master y lo subimos a master

- **git checkout master**
- **git merge feat_pojo**
- **git push**

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (feat_pojo)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git merge feat_pojo
Updating f691c9c..6a22655
Fast-forward
 src/Empleado.java | 18 ++++++++-----
 src/Main.java     |  2 ++
 2 files changed, 17 insertions(+), 3 deletions(-)

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/andvigofp/ejercicioGit4.git
 f691c9c..6a22655 master -> master
```

14 -Realizamos otras mejoras en la rama bug_deprecated_date_type para solucionar los problemas de la librería obsoleta java.util.Date, y los mergeamos.

- Creamos la última rama con **bug_deprecated_date_type** y nos cambiamos a la rama que creamos
 - **git branch bug_deprecated_date_type** o **git checkout -b bug_deprecated_date_type**
 - **git checkout bug_deprecated_date_type**

```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git branch bug_deprecated_date_type

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
master)
$ git checkout bug_deprecated_date_type
Switched to branch 'bug_deprecated_date_type'
```

- Modificamos el Main.java, lo dejamos igual que el commit “mostrando los problemas de Date”

```
src/Main.java
@@ -26,6 +26,14 @@ public static void main(String[] args) {
26      System.out.println("Si quiero mostrar solo el nombre: " + p.getNombre());
27      System.out.println("Si quiero mostrar solo la edad: " + p.getEdad());
28
29 +      /* Aunque la variable fechaContratación es immutable (final), su tipo (Date) es mutable, de modo que
30 +      todavía podría modificarse así: */
31 +      p.getFechaContratacion().setYear(10);
32 +
33 +      /* Además, Date es una clase de la primera versión de Java que solo almacena dos dígitos para años, de modo
34 +      que al ponerle el año 10 pensando en 2010, nos encontraremos que no sabemos si se refiere a 1910. */
35 +      System.out.println("Nueva fecha de contratación: " + p.getFechaContratacion().getYear());
36 +
29      37
```

- git status
- git add .
- git commit -m “mostrando el problema de Date”

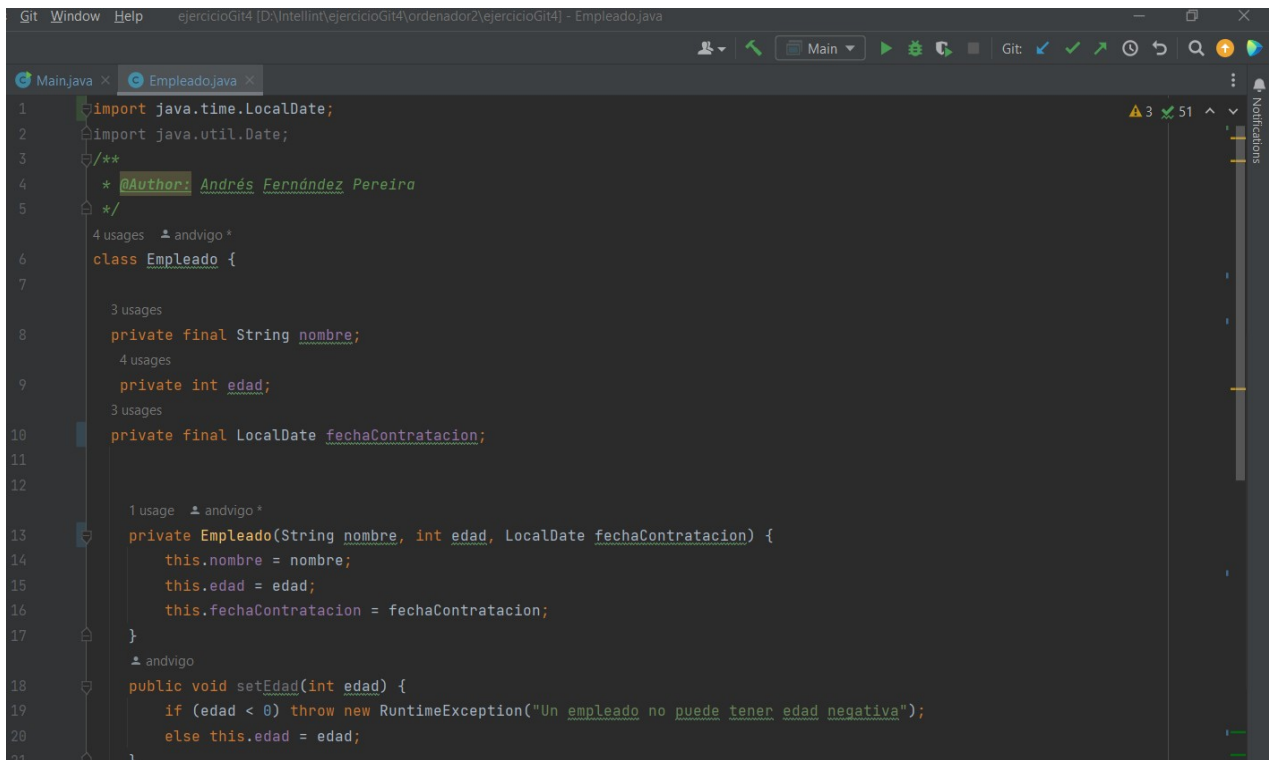
```
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (
bug_deprecated_date_type)
$ git status
On branch bug_deprecated_date_type
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")

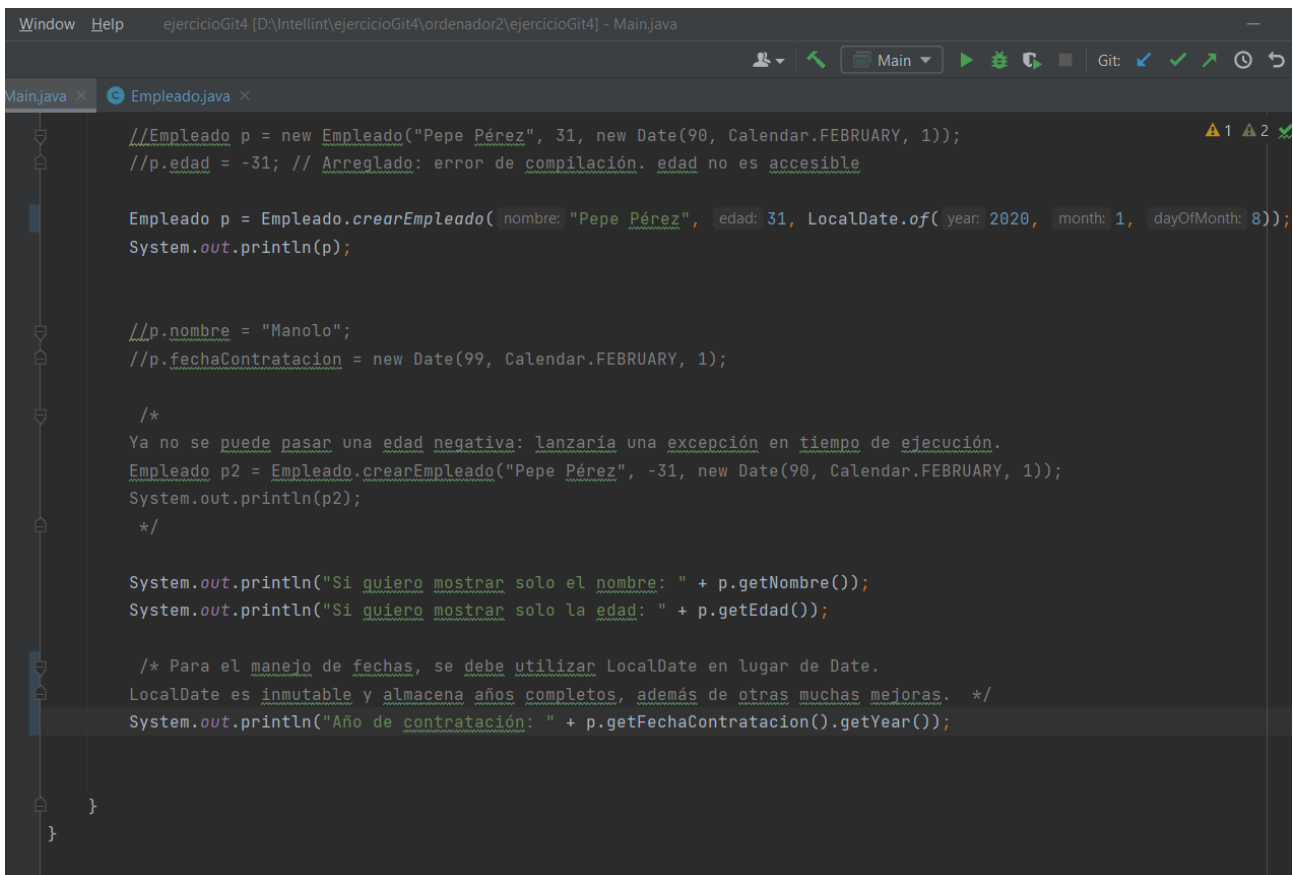
andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_deprecated_date_type)
$ git add .

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_deprecated_date_type)
$ git commit -m "mostrando los problemas de Date"
[bug_deprecated_date_type d1904df] mostrando los problemas de Date
1 file changed, 8 insertions(+)
```

- Modificamos el Main.java, Empleados.Java. Lo dejamos igual que en el commit “**migrado a LocalDate**”



```
1 import java.time.LocalDate;
2 import java.util.Date;
3 /**
4  * @Author: Andrés Fernández Pereira
5  */
6 class Empleado {
7
8     private final String nombre;
9     private int edad;
10    private final LocalDate fechaContratacion;
11
12    private Empleado(String nombre, int edad, LocalDate fechaContratacion) {
13        this.nombre = nombre;
14        this.edad = edad;
15        this.fechaContratacion = fechaContratacion;
16    }
17
18    public void setEdad(int edad) {
19        if (edad < 0) throw new RuntimeException("Un empleado no puede tener edad negativa");
20        else this.edad = edad;
21    }
22 }
```



```
1 //Empleado p = new Empleado("Pepe Pérez", 31, new Date(90, Calendar.FEBRUARY, 1));
2 //p.edad = -31; // Arreglado: error de compilación. edad no es accesible
3
4 Empleado p = Empleado.crearEmpleado( nombre: "Pepe Pérez", edad: 31, LocalDate.of( year: 2020, month: 1, dayOfMonth: 8));
5 System.out.println(p);
6
7 //p.nombre = "Manolo";
8 //p.fechaContratacion = new Date(99, Calendar.FEBRUARY, 1);
9
10 /*
11 Ya no se puede pasar una edad negativa: lanzaría una excepción en tiempo de ejecución.
12 Empleado p2 = Empleado.crearEmpleado("Pepe Pérez", -31, new Date(90, Calendar.FEBRUARY, 1));
13 System.out.println(p2);
14 */
15
16 System.out.println("Si quiero mostrar solo el nombre: " + p.getNombre());
17 System.out.println("Si quiero mostrar solo la edad: " + p.getEdad());
18
19 /* Para el manejo de fechas, se debe utilizar LocalDate en lugar de Date.
20 LocalDate es inmutable y almacena años completos, además de otras muchas mejoras. */
21 System.out.println("Año de contratación: " + p.getFechaContratacion().getYear());
22
23 }
```


- **git status**
- **git add .**
- **git commit -m “migrando a LocalDate”**

```

1 file changed, 0 insertions(+)

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_deprecated_date_type)
$ git status
On branch bug_deprecated_date_type
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/Empleado.java
        modified:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_deprecated_date_type)
$ git add .

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_deprecated_date_type)
$ git commit -m "migrado a LocalDate"
[bug_deprecated_date_type a25177b] migrado a LocalDate
2 files changed, 10 insertions(+), 14 deletions(-)

```

- Ahora toca subir el contenido a la rama **bug_deprecated_date_type**

- **git push --set-upstream origin bug_deprecated_date_type**

```

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_deprecated_date_type)
$ git push --set-upstream origin bug_deprecated_date_type
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 1.27 KiB | 1.27 MiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 3 local objects.
remote:
remote: Create a pull request for 'bug_deprecated_date_type' on GitHub by visiting:
remote:   https://github.com/andvigofp/ejercicioGit4/pull/new/bug_deprecated_date_type
remote:
To https://github.com/andvigofp/ejercicioGit4.git
 * [new branch]      bug_deprecated_date_type -> bug_deprecated_date_type
branch 'bug_deprecated_date_type' set up to track 'origin/bug_deprecated_date_type'.

```

- Ultimo quedaría pasar a la rama master y mergear la juntar la ramas y subir los commits a la rama master

- **git checkout master**
- **git merge bug_deprecated_date_type**
- **git push**

```

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (bug_deprecated_date_type)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git merge bug_deprecated_date_type
Updating 6a22655..a25177b
Fast-forward
 src/Empleado.java | 9 ++++----
 src/Main.java     | 9 ++++++--
 2 files changed, 11 insertions(+), 7 deletions(-)

andre@MSI_Portatil MINGW64 /d/Intellint/ejercicioGit4/ordenador2/ejercicioGit4 (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/andvigofp/ejercicioGit4.git
 6a22655..a25177b master -> master

```