

## Actividad 4 - Utilizando sistemas de control de versiones

Brayan Steven Bonilla Castellanos

Juan Carlos Monsalve Gómez

Corporación Universitaria Iberoamericana

Ingeniería de Software

Mantenimiento de Software

## CONTENIDO

1	ENLACE GIT	3
2	DESCRIPCIÓN DEL CONTEXTO	4
3	FLUJO DE TRABAJO INICIAL	5
4	HISTORIAS DE USUARIO	9
4.1	Historia de Usuario 1: API servicios desarrollados	9
4.2	Historia de usuario 2: Servicio REST tipo POST	12
4.3	Historia de usuario 3: vista para registro de horas	17
4.4	Historia de usuario 4: servicio REST tipo GET	20
4.5	Historia de usuario 5: vista consulta horas	25
5	DOCUMENTACIÓN DE COMANDOS GIT USADOS	29
5.1	git clone	29
5.2	git status	29
5.3	git add .	29
5.4	git commit -m	30
5.5	git pull	30
5.6	git push	30
5.7	git checkout	31
5.8	git merge	31
5.9	git branch	31
5.10	git log	31
6	REFERENCIAS	32

# 1 Enlace Git

<https://github.com/BrayanBonilla-Ibero/HorasLaborales>

You now have push access to the BrayanBonilla-Ibero/HorasLaborales repository.

BrayanBonilla-Ibero / HorasLaborales Public

Unwatch 2 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags

Go to file Add file Code

About

Proyecto desarrollado con Spring Boot y Angular

Readme 0 stars 2 watching 0 forks

Releases

No releases published. Create a new release

Packages

No packages published. Publish your first package

Languages

Java 59.1% CSS 18.4% TypeScript 17.4% HTML 2.7% JavaScript 2.4%

BrayanBonilla-Ibero Documento base 7a7a3c8 9 hours ago 6 commits

Backup DB	Incorporacion del proyecto base	10 hours ago
HandyMan	Solución HU Servicio Rest Get	10 hours ago
handyman-fe	Incorporacion del proyecto base	10 hours ago
Actividad.docx	Documento base	9 hours ago
README.md	Initial commit	11 hours ago

README.md

## HorasLaborales

Proyecto desarrollado con Spring Boot y Angular

## **2 Descripción del contexto**

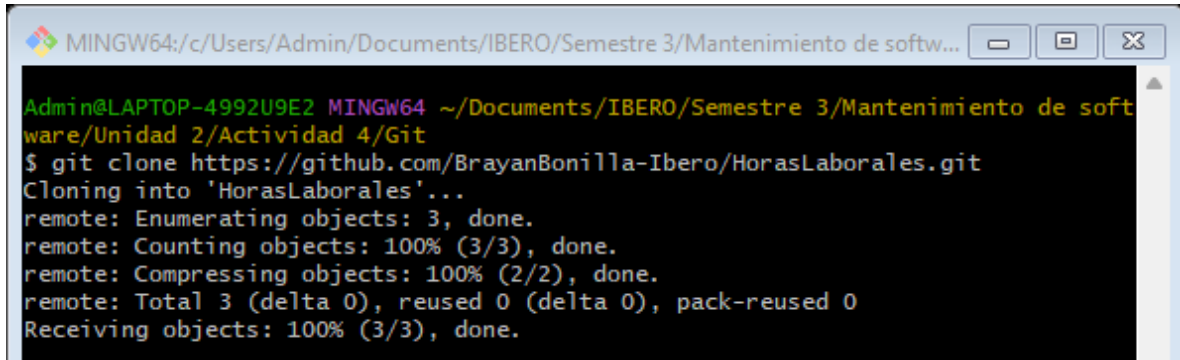
Se requiere implementar un sistema de información que permita el registro de horas laboradas por cada trabajador (Horas diurnas, horas nocturnas, horas dominicales, horas diurnas extras, horas nocturnas extras, horas dominicales extras).

El sistema es elaborado por un equipo de trabajo compuesto por dos desarrolladores, se utiliza el software GIT como sistema de control de versiones y todo el código fuente de la aplicación aparece almacenado en GitHub.

La aplicación se desarrolla en el Backend con Java con Spring Boot y en el FrontEnd con Angular v14.

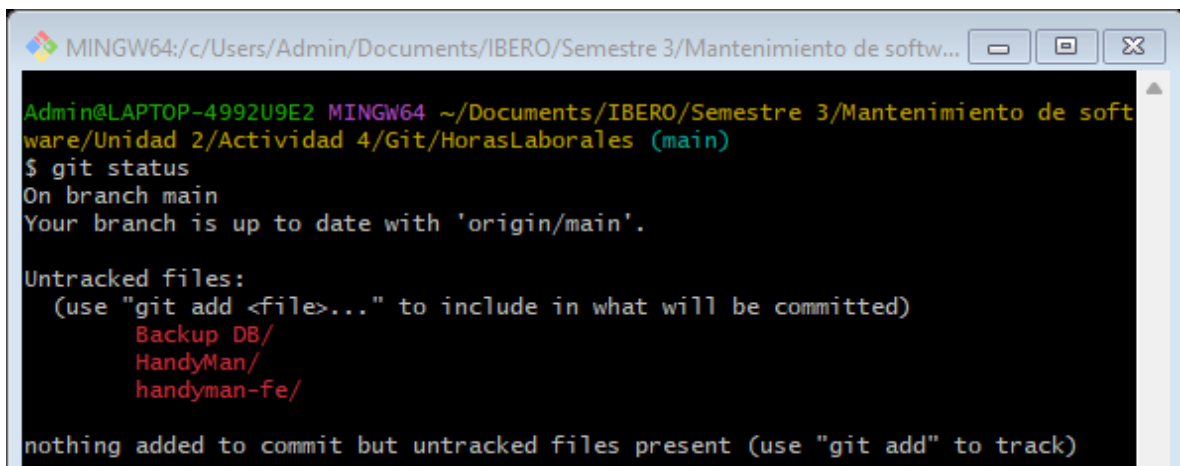
### 3 Flujo de trabajo inicial

Ingresamos el comando **git clone**, para clonar el repositorio remoto al equipo local.



```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git clone https://github.com/BrayanBonilla-Ibero/HorasLaborales.git
Cloning into 'HorasLaborales'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

Una vez clonado el repositorio, incorporamos el proyecto base y verificamos los cambios realizados con el comando **git status**



```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ 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)
    Backup DB/
    HandyMan/
    handyman-fe/

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

Procedemos a agregar los cambios realizados a un stage con el comando **git add**.

```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git add .
warning: in the working copy of 'HandyMan/.gitignore', LF will be replaced by CR
LF the next time Git touches it
warning: in the working copy of 'HandyMan/.mvn/wrapper/maven-wrapper.properties'
, LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'HandyMan/mvnw', LF will be replaced by CRLF the
next time Git touches it
warning: in the working copy of 'HandyMan/mvnw.cmd', LF will be replaced by CRLF
the next time Git touches it
warning: in the working copy of 'HandyMan/pom.xml', LF will be replaced by CRLF
the next time Git touches it
warning: in the working copy of 'HandyMan/src/main/java/com/example/handyMan/Han
dyManApplication.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'HandyMan/src/test/java/com/example/handyMan/Han
dyManApplicationTests.java', LF will be replaced by CRLF the next time Git touch
es it
warning: in the working copy of 'handyman-fe/.browserslistrc', LF will be replac
ed by CRLF the next time Git touches it
warning: in the working copy of 'handyman-fe/.editorconfig', LF will be replaced
by CRLF the next time Git touches it
warning: in the working copy of 'handyman-fe/.gitignore', LF will be replaced by
```

Una vez confirmado los cambios procedemos a realizar un commit, con el comando **git commit -m**

```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git commit -m "Incorporacion del proyecto base"
[main 0c90e34] Incorporacion del proyecto base
 82 files changed, 23389 insertions(+)
 create mode 100644 Backup DB/backup.backup
 create mode 100644 HandyMan/.gitignore
 create mode 100644 HandyMan/.mvn/wrapper/maven-wrapper.jar
 create mode 100644 HandyMan/.mvn/wrapper/maven-wrapper.properties
 create mode 100644 HandyMan/mvnw
 create mode 100644 HandyMan/mvnw.cmd
 create mode 100644 HandyMan/pom.xml
 create mode 100644 HandyMan/src/main/java/com/example/handyMan/HandyManApplicat
ion.java
 create mode 100644 HandyMan/src/main/java/com/example/handyMan/application/doma
in/Report.java
 create mode 100644 HandyMan/src/main/java/com/example/handyMan/application/port
s/in/ReportUseCase.java
 create mode 100644 HandyMan/src/main/java/com/example/handyMan/application/port
s/in/WeekCalculatorUseCase.java
 create mode 100644 HandyMan/src/main/java/com/example/handyMan/application/port
s/out/ReportRepository.java
 create mode 100644 HandyMan/src/main/java/com/example/handyMan/application/port
```

Verificamos que el commit se registrara con éxito, con el comando **git status** y debe registrar 1 commit pendiente por subir

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
create mode 100644 handyman-fe/src/environments/environment.prod.ts
create mode 100644 handyman-fe/src/environments/environment.ts
create mode 100644 handyman-fe/src/favicon.ico
create mode 100644 handyman-fe/src/index.html
create mode 100644 handyman-fe/src/main.ts
create mode 100644 handyman-fe/src/polyfills.ts
create mode 100644 handyman-fe/src/styles.css
create mode 100644 handyman-fe/src/test.ts
create mode 100644 handyman-fe/tsconfig.app.json
create mode 100644 handyman-fe/tsconfig.json
create mode 100644 handyman-fe/tsconfig.spec.json

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$
```

Verificamos que no tengamos cambios pendientes por obtener con el comando **git pull**

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git pull
Already up to date.

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$
```

Procedemos a subir los cambios al repositorio remoto en la rama principal, con el comando **git push**

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git push
Enumerating objects: 132, done.
Counting objects: 100% (132/132), done.
Delta compression using up to 16 threads
Compressing objects: 100% (114/114), done.
Writing objects: 100% (131/131), 280.45 KiB | 6.23 MiB/s, done.
Total 131 (delta 12), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (12/12), done.
To https://github.com/BrayanBonilla-Ibero/HorasLaborales.git
   af24e56..0c90e34  main -> main

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ .....
```

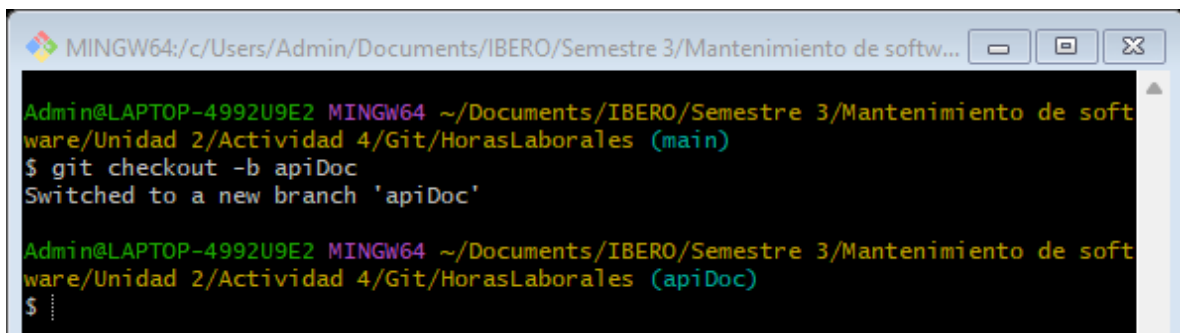


## 4 Historias de usuario

### 4.1 Historia de Usuario 1: API servicios desarrollados

Se requiere la construcción de un api que contenga la documentación de los servicios desarrollados.

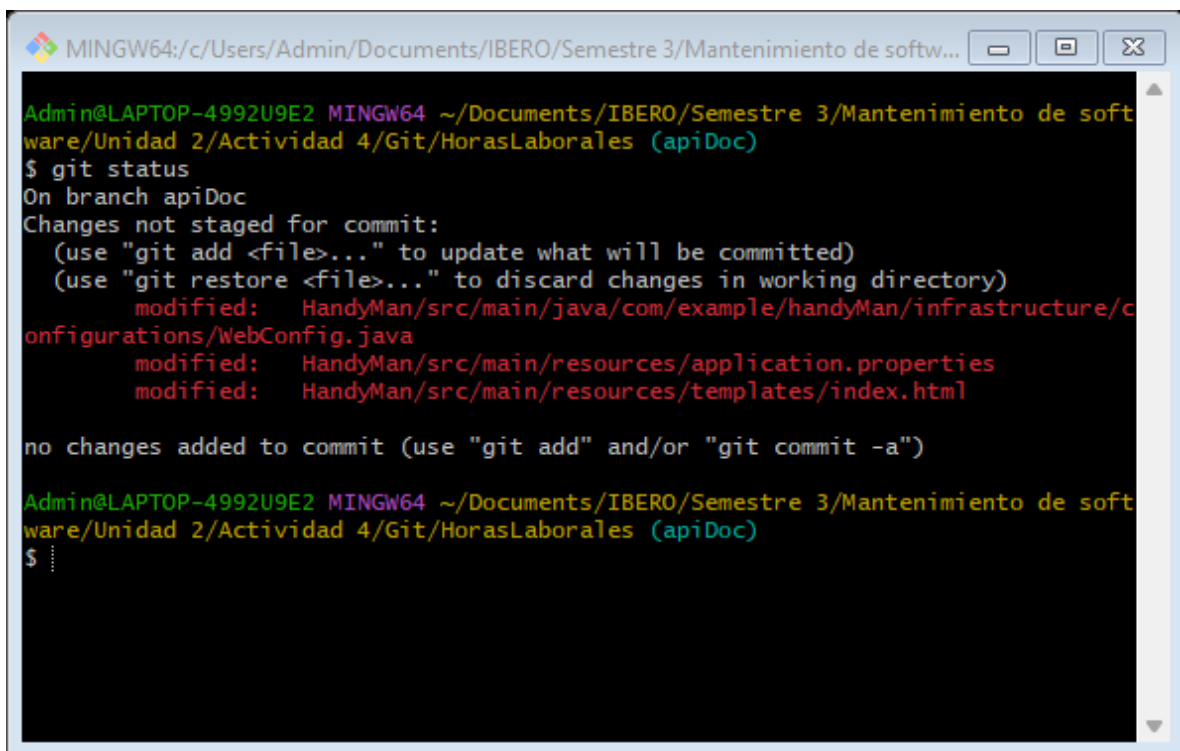
Para llevar a cabo esta tarea vamos a crear una rama específica para poder trabajar esta HU, con el comando **git checkout -b apiDoc**



```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git checkout -b apiDoc
Switched to a new branch 'apiDoc'

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$
```

Una vez se trabaja la HU en el código fuente, procedemos a verificar los cambios, con el comando **git status**

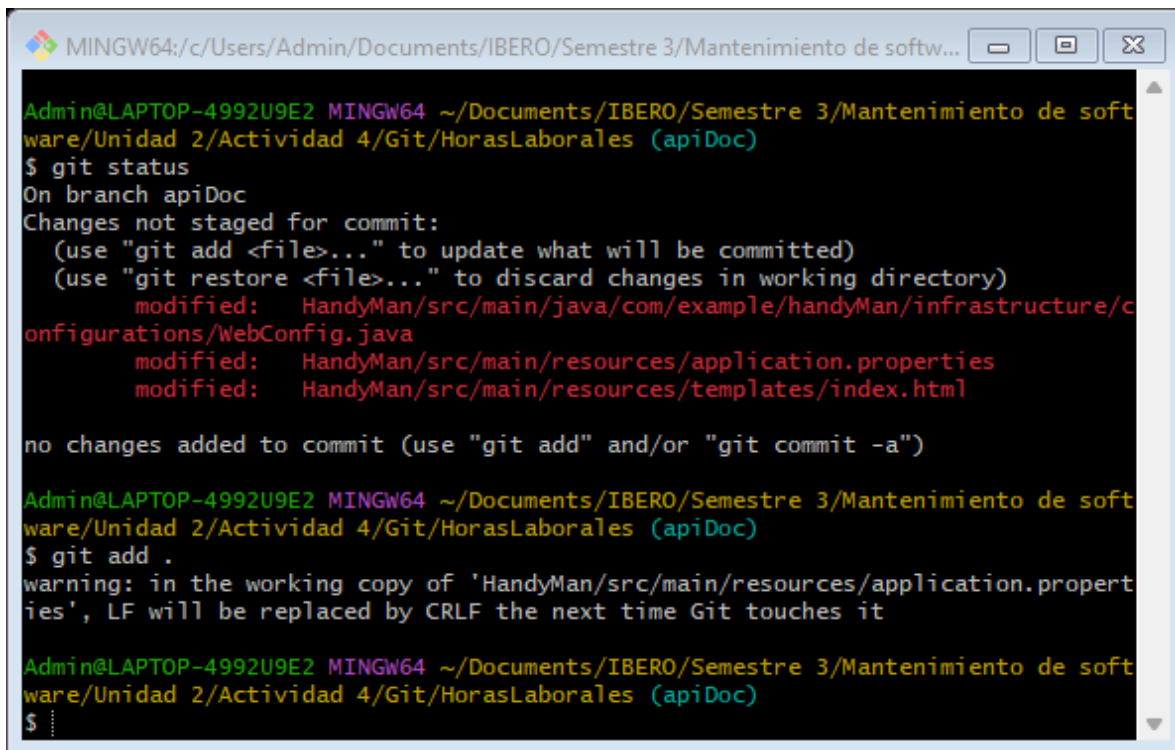


```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git status
On branch apiDoc
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   HandyMan/src/main/java/com/example/handyMan/infrastructure/c
onfigurations/WebConfig.java
    modified:   HandyMan/src/main/resources/application.properties
    modified:   HandyMan/src/main/resources/templates/index.html

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

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$
```

Agregamos los cambios al stage, con el comando **git add** .



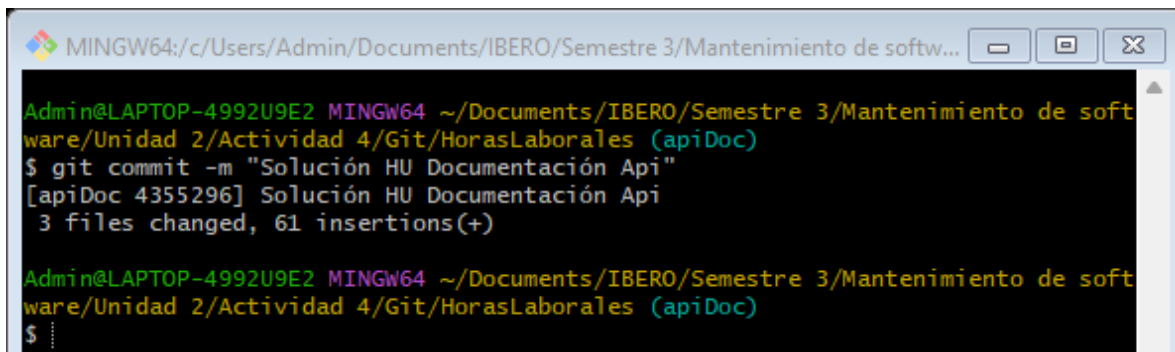
```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
ware/Unidad 2/Actividad 4/Git/HorasLaborales (apiDoc)
$ git status
On branch apiDoc
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   HandyMan/src/main/java/com/example/handyMan/infrastructure/c
onfigurations/WebConfig.java
    modified:   HandyMan/src/main/resources/application.properties
    modified:   HandyMan/src/main/resources/templates/index.html

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

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (apiDoc)
$ git add .
warning: in the working copy of 'HandyMan/src/main/resources/application.propert
ies', LF will be replaced by CRLF the next time Git touches it

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (apiDoc)
$
```

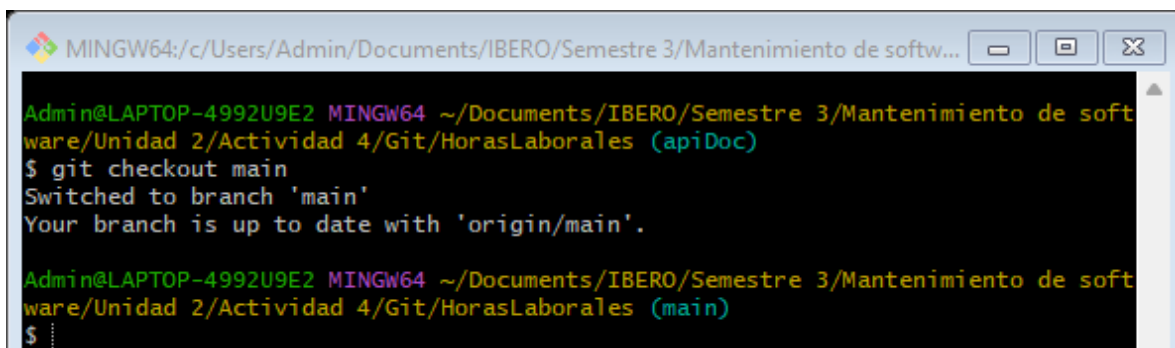
Realizamos el commit con los cambios realizados con el siguiente comando



```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
ware/Unidad 2/Actividad 4/Git/HorasLaborales (apiDoc)
$ git commit -m "Solución HU Documentación Api"
[apiDoc 4355296] Solución HU Documentación Api
3 files changed, 61 insertions(+)

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (apiDoc)
$
```

Una vez realizado el commit, procedemos a pasar a la rama principal con el siguiente comando



```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
ware/Unidad 2/Actividad 4/Git/HorasLaborales (apiDoc)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$
```

Una vez verificada la solución a la historia de usuario, procedemos a realizar el merge con la rama principal, con el siguiente comando

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git merge apiDoc
Updating 0c90e34..4355296
Fast-forward
 ../infrastructure/configurations/WebConfig.java | 19 ++++++++
 HandyMan/src/main/resources/application.properties | 9 +++++
 HandyMan/src/main/resources/templates/index.html | 33 ++++++++
 3 files changed, 61 insertions(+)

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ .....
```

Subimos los cambios incorporados a la rama principal al repositorio remoto, con el siguiente comando

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git push
Enumerating objects: 31, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (16/16), 1.98 KiB | 507.00 KiB/s, done.
Total 16 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/BrayanBonilla-Ibero/HorasLaborales.git
 0c90e34..4355296 main -> main

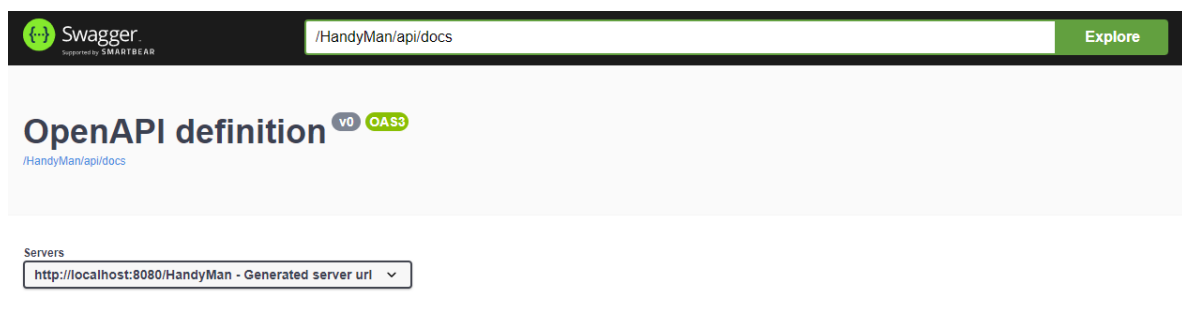
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ .....
```

Como ya nos vamos a utilizar más la rama creada para trabajar la HU, procedemos a borrarla con el siguiente comando

```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git branch -d apiDoc
Deleted branch apiDoc (was 4355296).

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ .....
```

El resultado obtenido es el siguiente



## 4.2 Historia de usuario 2: Servicio REST tipo POST

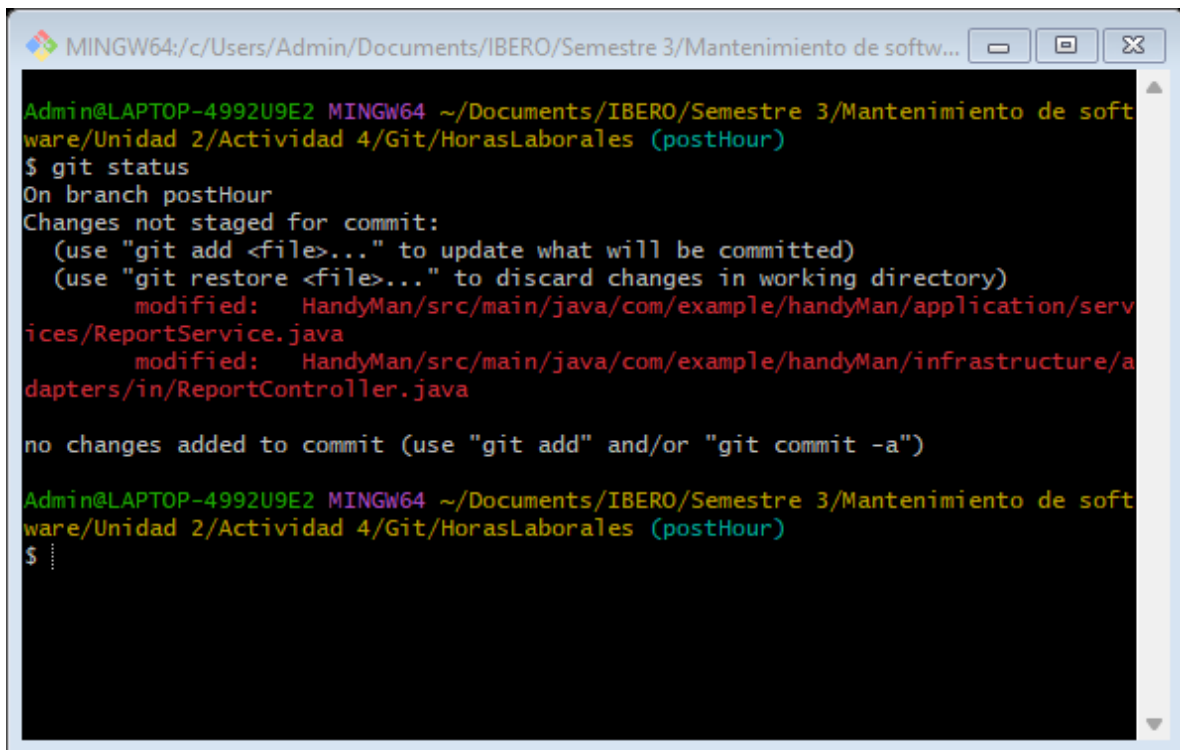
Se requiere crear un servicio Rest de tipo Post que permita guardar el registro de las horas por trabajador.

Para llevar a cabo esta tarea vamos a crear una rama específica para poder trabajar esta HU, con el comando **git checkout -b postHour**

```
MINGW64:/c:/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git checkout -b postHour
Switched to a new branch 'postHour'

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (postHour)
$ .....
```

Una vez trabaja la HU en el código fuente, procedemos a verificar los cambios, con el comando **git status**

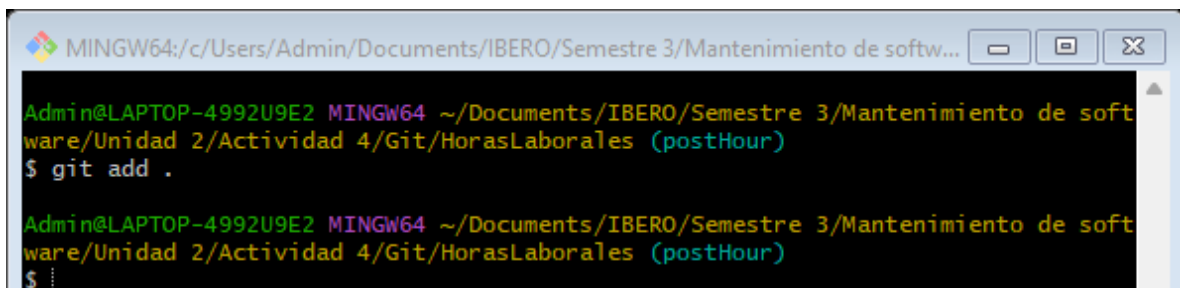
A terminal window titled 'MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...' showing the output of the 'git status' command. The output indicates that the user is on the 'postHour' branch and that there are changes not staged for commit. The changes are listed as 'modified' files: 'HandyMan/src/main/java/com/example/handyMan/application/services/ReportService.java' and 'HandyMan/src/main/java/com/example/handyMan/infrastructure/adapters/in/ReportController.java'. The terminal also shows instructions on how to use 'git add' and 'git restore' to manage these changes.

```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (postHour)
$ git status
On branch postHour
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:   HandyMan/src/main/java/com/example/handyMan/application/serv
ices/ReportService.java
        modified:   HandyMan/src/main/java/com/example/handyMan/infrastructure/a
dapters/in/ReportController.java

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

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (postHour)
$ :
```

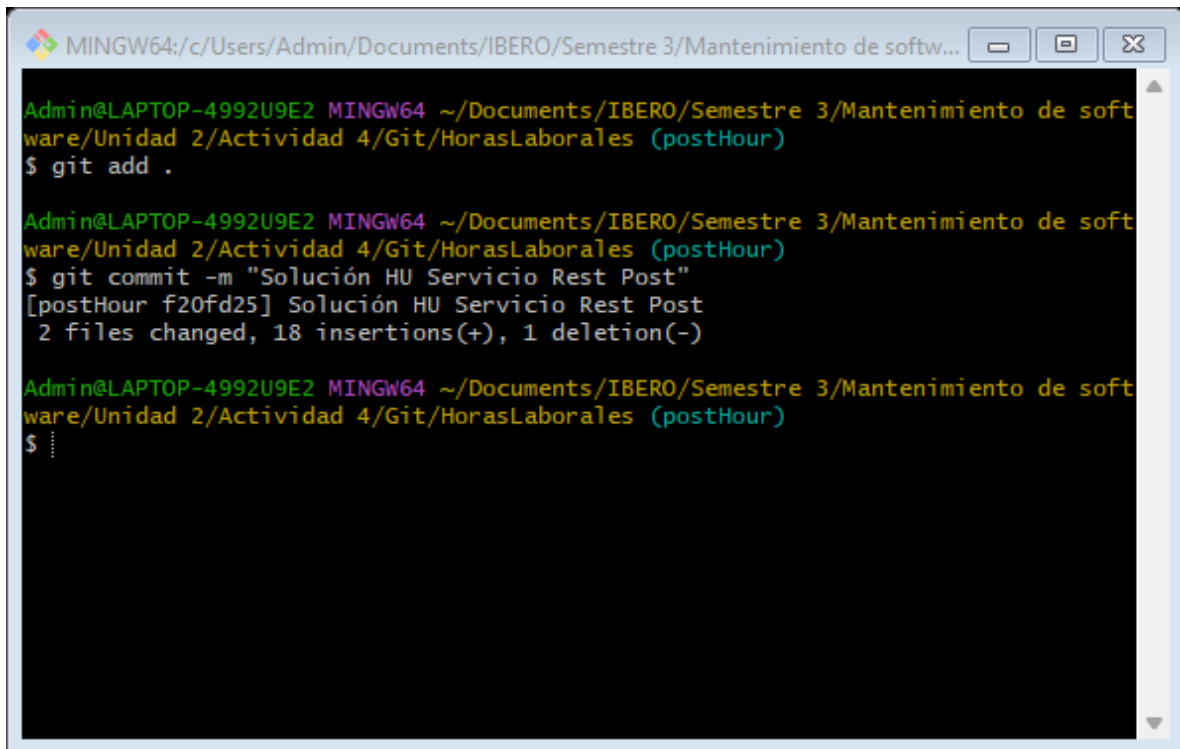
Agregamos los cambios al stage, con el comando **git add** .

A terminal window titled 'MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...' showing the execution of the 'git add .' command. The prompt '\$ git add .' is shown, followed by a new prompt '\$ :'.

```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (postHour)
$ git add .

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (postHour)
$ :
```

Realizamos el commit con los cambios realizados con el siguiente comando

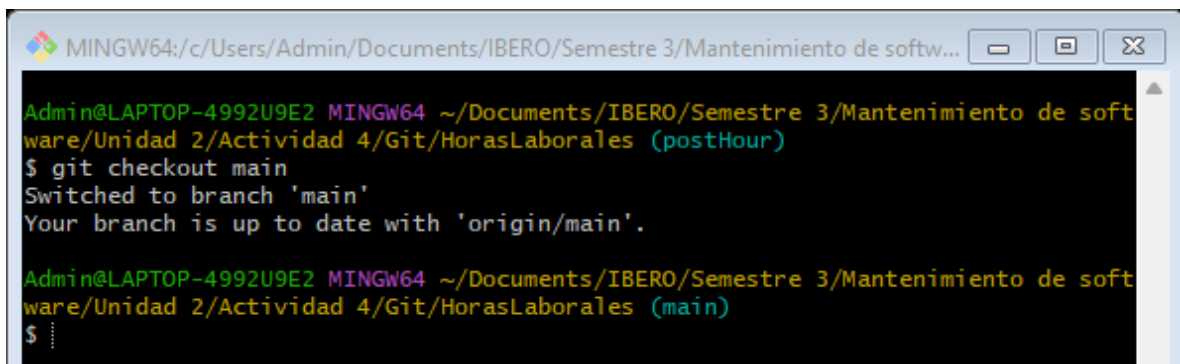
A terminal window titled 'MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...' showing a series of git commands. The user is in a directory path that includes 'HorasLaborales (postHour)'. They run 'git add .' and then 'git commit -m "Solución HU Servicio Rest Post"'. The commit message is 'Solución HU Servicio Rest Post' and the commit hash is 'f20fd25'. The commit shows '2 files changed, 18 insertions(+), 1 deletion(-)'. The prompt returns to the shell.

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (postHour)
$ git add .

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (postHour)
$ git commit -m "Solución HU Servicio Rest Post"
[postHour f20fd25] Solución HU Servicio Rest Post
 2 files changed, 18 insertions(+), 1 deletion(-)

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (postHour)
$ .....
```

Una vez realizado el commit, procedemos a pasar a la rama principal con el siguiente comando

A terminal window titled 'MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...' showing the 'git checkout main' command. The output indicates that the user has switched to the 'main' branch and that their branch is up to date with 'origin/main'. The prompt returns to the shell with '(main)' in parentheses.

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (postHour)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ .....
```

Una vez verificada la solución a la historia de usuario, procedemos a realizar el merge con la rama principal, con el siguiente comando

```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git merge postHour
Updating 4355296..f20fd25
Fast-forward
 .../example/handyMan/application/services/ReportService.java | 11 ++++++++
 .../handyMan/infrastructure/adapters/in/ReportController.java |  8 ++++++
 2 files changed, 18 insertions(+), 1 deletion(-)

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$
```

Subimos los cambios incorporados a la rama principal al repositorio remoto, con el siguiente comando

```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git push
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (16/16), 1.59 KiB | 541.00 KiB/s, done.
Total 16 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), completed with 6 local objects.
To https://github.com/BrayanBonilla-Ibero/HorasLaborales.git
 4355296..f20fd25  main -> main

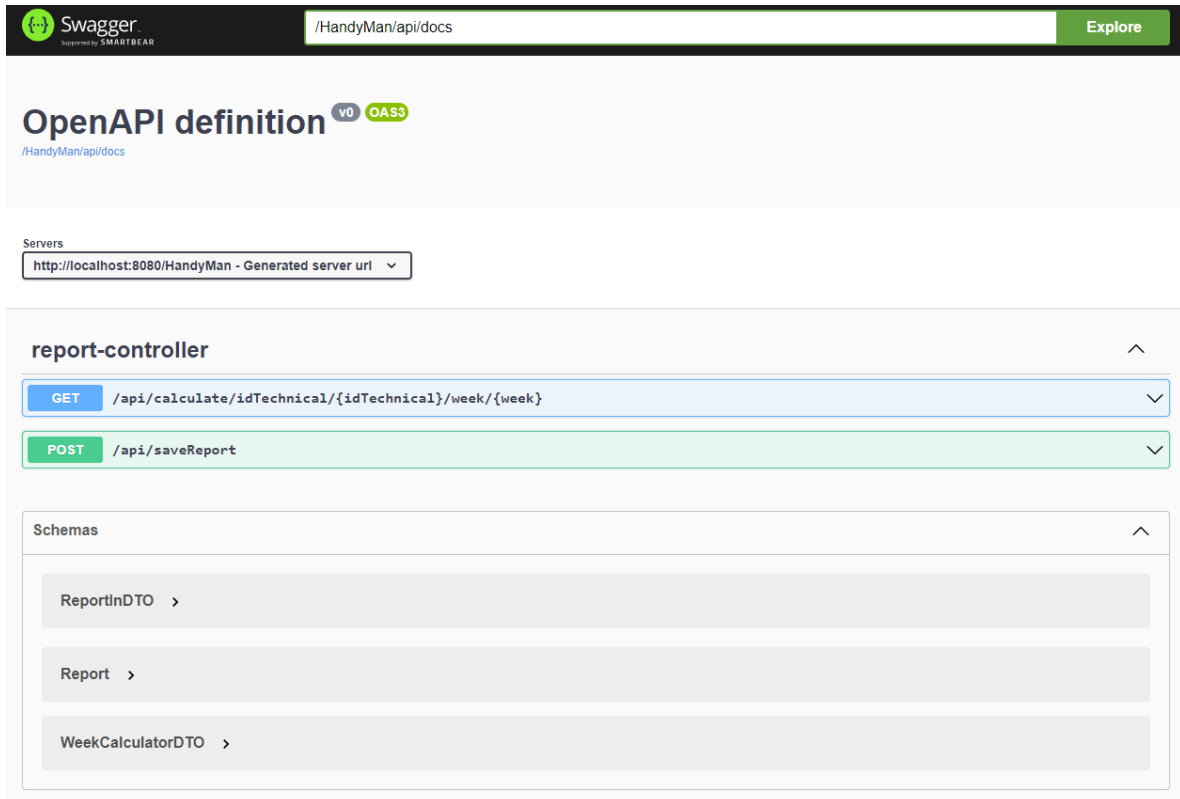
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$
```

Como ya nos vamos a utilizar más la rama creada para trabajar la HU, procedemos a borrarla con el siguiente comando

```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git branch -d postHour
Deleted branch postHour (was f20fd25).

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$
```

El resultado obtenido es el siguiente



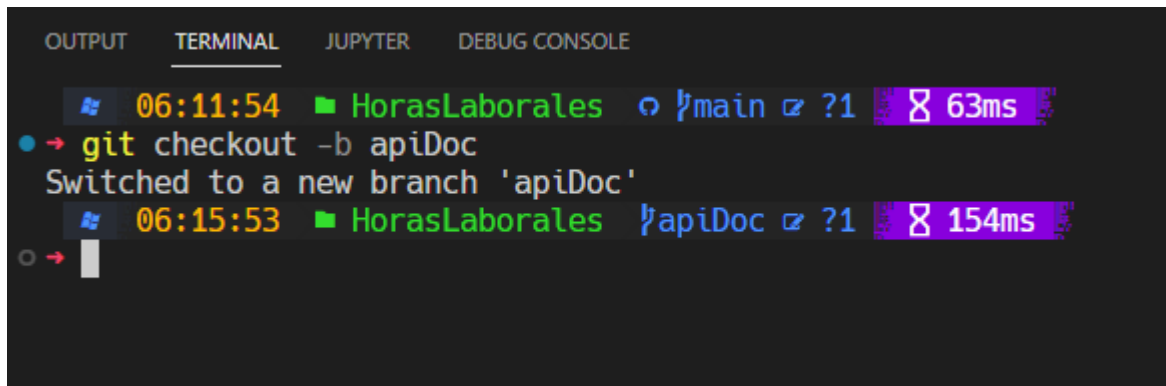
The image shows the Swagger UI for an API. At the top, there's a Swagger logo and a search bar containing "/HandyMan/api/docs". Below this, the text "OpenAPI definition" is displayed with version "v0" and "OAS3" tags. A "Servers" section shows a dropdown menu with "http://localhost:8080/HandyMan - Generated server url". The main section is titled "report-controller" and contains two endpoints: a GET endpoint "/api/calculate/idTechnical/{idTechnical}/week/{week}" and a POST endpoint "/api/saveReport". Below the endpoints, there's a "Schemas" section with three entries: "ReportInDTO", "Report", and "WeekCalculatorDTO", each with a right-pointing arrow.



### 4.3 Historia de usuario 3: vista para registro de horas

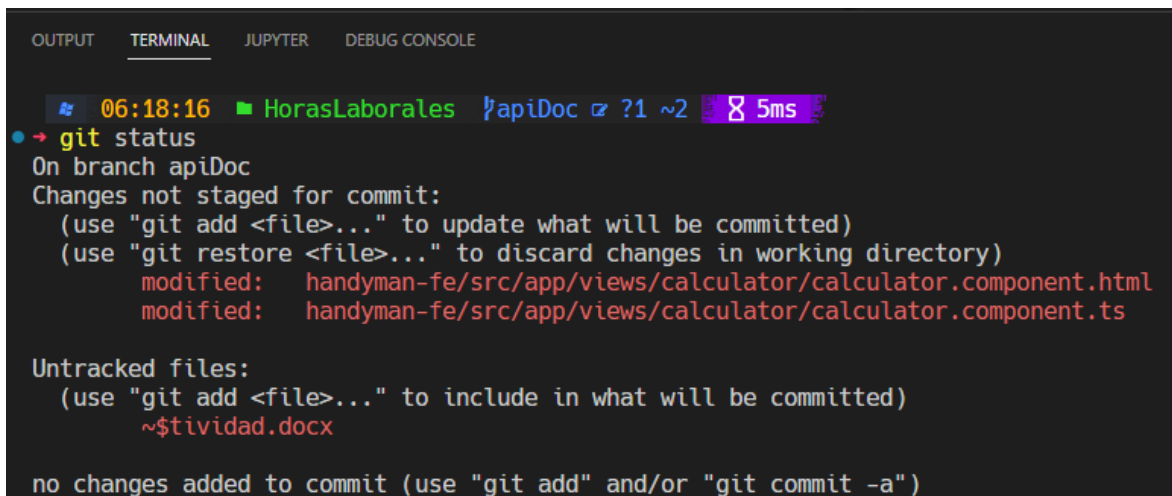
Se requiere diseñar la vista al usuario que le permita registrar sus horas laboradas.

Para llevar a cabo esta tarea vamos a crear una rama específica para poder trabajar esta hu, con el comando **git checkout -b apiDoc**



```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE
06:11:54  HorasLaborales  ?main ?1  63ms
git checkout -b apiDoc
Switched to a new branch 'apiDoc'
06:15:53  HorasLaborales  ?apiDoc ?1  154ms
```

Una vez trabaja la HU en el código fuente, procedemos a verificar los cambios, con el comando **git status**

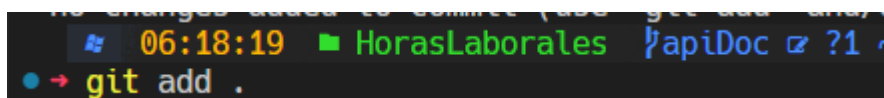


```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE
06:18:16  HorasLaborales  ?apiDoc ?1 ~2  5ms
git status
On branch apiDoc
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   handyman-fe/src/app/views/calculator/calculator.component.html
    modified:   handyman-fe/src/app/views/calculator/calculator.component.ts

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ~$tividad.docx

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

Agregamos los cambios al stage, con el comando **git add .**



```
06:18:19  HorasLaborales  ?apiDoc ?1 ~
git add .
```

Realizamos el commit con los cambios realizados con el siguiente comando

```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

06:19:06  ■ HorasLaborales  ?apiDoc  +1 ~2  8 4ms
• → git commit -m "vista de usuario para registro de horas"
[apiDoc 409ac23] vista de usuario para registro de horas
3 files changed, 94 insertions(+)
create mode 100644 ~$tividad.docx
06:19:19  ■ HorasLaborales  ?apiDoc  8 291ms
○ →
```

Una vez realizado el commit, procedemos a pasar a la rama principal con el siguiente comando

```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

06:20:08  ■ HorasLaborales  ?apiDoc  ?1 ~1  8 11ms
• → git checkout main
Switched to branch 'main'
M      Actividad.docx
Your branch is up to date with 'origin/main'.
06:20:14  ■ HorasLaborales  ?main  ?1 ~1  8 212ms
○ →
```

Una vez verificada la solución a la historia de usuario, procedemos a realizar el merge con la rama principal, con el siguiente comando

```
06:20:14  ■ HorasLaborales  ?main  ?1 ~1  8 212ms
• → git merge apiDoc
Updating 7a7a3c9..409ac23
Fast-forward
.../app/views/calculator/calculator.component.html | 58 ++++++
.../app/views/calculator/calculator.component.ts   | 36 ++++++
~$tividad.docx                                       | Bin 0 -> 162 bytes
3 files changed, 94 insertions(+)
create mode 100644 ~$tividad.docx
06:21:02  ■ HorasLaborales  ?main  ?1 ~1  8 315ms
○ →
```

Subimos los cambios incorporados a la rama principal al repositorio remoto, con el siguiente comando

```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

06:21:31  ■ HorasLaborales  ○ |main  21 ~1  8 4ms
● → git push
Enumerating objects: 18, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.81 KiB | 618.00 KiB/s, done.
Total 10 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/BrayanBonilla-Ibero/HorasLaborales.git
7a7a3c9..409ac23  main -> main
06:21:37  ■ HorasLaborales  ○ |main  21 ~1  8 3.853s
○ →
```

Como ya nos vamos a utilizar más la rama creada para trabajar la HU, procedemos a borrarla con el siguiente comando

```
06:21:37  ■ HorasLaborales  ○ |main  21 ~1  8 3.853s
● → git branch -d apiDoc
Deleted branch apiDoc (was 409ac23).
06:22:09  ■ HorasLaborales  ○ |main  21 ~1  8 173ms
○ →
```

El resultado obtenido es el siguiente



Registro Reporte de Servicio

Identificación Técnico

Técnico

Identificación Servicio

Servicio

Fecha Inicio

dd/mm/aaaa --:-- -----

Fecha Fin

dd/mm/aaaa --:-- -----

GUARDAR

CALCULADORA DE HORAS >

#### 4.4 Historia de usuario 4: servicio REST tipo GET

Se requiere crear un servicio Rest de tipo Get que permita consultar las horas laboradas de un trabajador en una semana determinada.

Para llevar a cabo esta tarea vamos a crear una rama específica para poder trabajar esta hu, con el comando git checkout -b gettHour

```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git checkout -b gettHour
Switched to a new branch 'gettHour'

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$
```

Una vez trabaja la HU en el código fuente, procedemos a verificar los cambios, con el comando git status

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git checkout -b gettHour
Switched to a new branch 'gettHour'

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$ git status
On branch gettHour
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:   HandyMan/src/main/java/com/example/handyMan/application/serv
ices/ReportService.java
    modified:   HandyMan/src/main/java/com/example/handyMan/infrastructure/a
dapters/in/ReportController.java

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

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$ .....
```

Agregamos los cambios al stage, con el comando git add .

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git checkout -b gettHour
Switched to a new branch 'gettHour'

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$ git status
On branch gettHour
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:   HandyMan/src/main/java/com/example/handyMan/application/serv
ices/ReportService.java
    modified:   HandyMan/src/main/java/com/example/handyMan/infrastructure/a
dapters/in/ReportController.java

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

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$ git add .

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$ .....
```

Realizamos el commit con los cambios realizados con el siguiente comando

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
On branch gettHour
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:   HandyMan/src/main/java/com/example/handyMan/application/services/ReportService.java
    modified:   HandyMan/src/main/java/com/example/handyMan/infrastructure/adapters/in/ReportController.java

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

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$ git add .

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$ git commit -m "Solución HU Servicio Rest Get"
[gettHour c43771c] Solución HU Servicio Rest Get
 2 files changed, 63 insertions(+), 1 deletion(-)

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$
```

Una vez realizado el commit, procedemos a pasar a la rama principal con el siguiente comando

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
    modified:   HandyMan/src/main/java/com/example/handyMan/infrastructure/adapters/in/ReportController.java

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

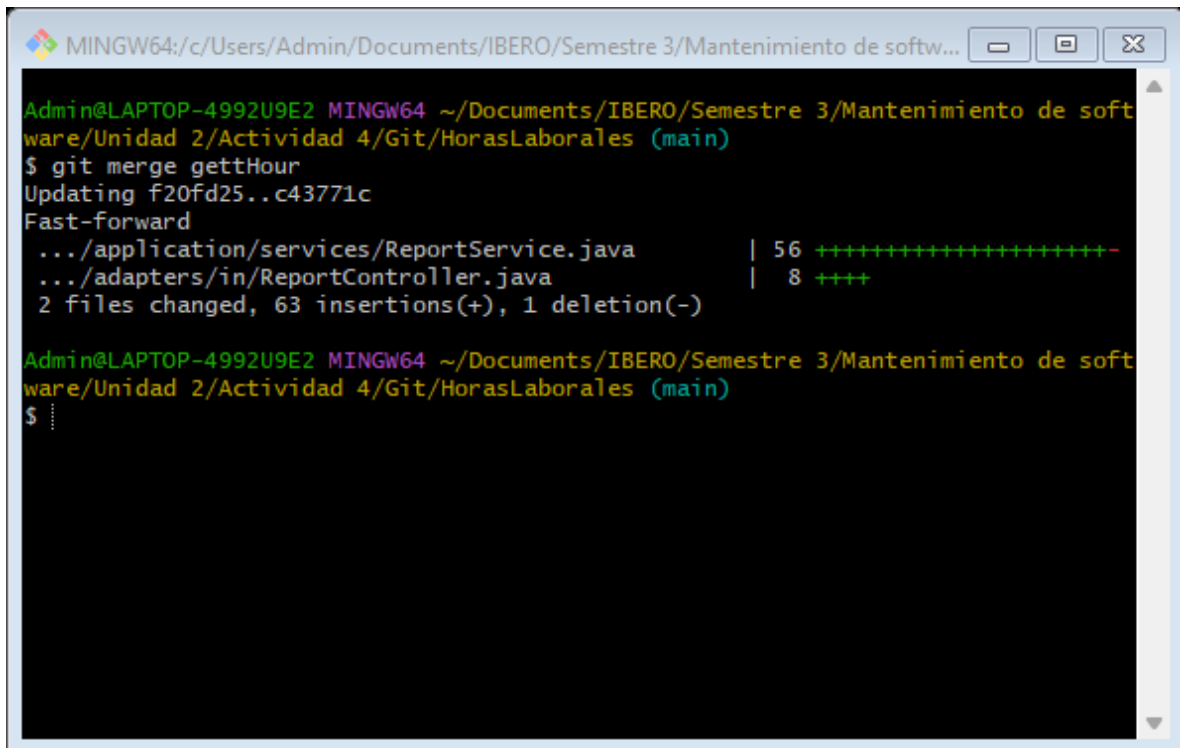
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$ git add .

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$ git commit -m "Solución HU Servicio Rest Get"
[gettHour c43771c] Solución HU Servicio Rest Get
 2 files changed, 63 insertions(+), 1 deletion(-)

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (gettHour)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de software/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$
```

Una vez verificada la solución a la historia de usuario, procedemos a realizar el merge con la rama principal, con el siguiente comando



```
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ git merge gettHour
Updating f20fd25..c43771c
Fast-forward
 .../application/services/ReportService.java      | 56 ++++++
 .../adapters/in/ReportController.java            |  8 +++
 2 files changed, 63 insertions(+), 1 deletion(-)

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de softw...
$ .....
```

Subimos los cambios incorporados a la rama principal al repositorio remoto, con el siguiente comando

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git merge gettHour
Updating f20fd25..c43771c
Fast-forward
 .../application/services/ReportService.java      | 56 ++++++
 .../adapters/in/ReportController.java            |  8 ++++
 2 files changed, 63 insertions(+), 1 deletion(-)

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git push
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (16/16), 2.21 KiB | 753.00 KiB/s, done.
Total 16 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/BrayanBonilla-Ibero/HorasLaborales.git
 f20fd25..c43771c  main -> main

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$
```

Como ya nos vamos a utilizar más la rama creada para trabajar la HU, procedemos a borrarla con el siguiente comando

```
MINGW64:/c/Users/Admin/Documents/IBERO/Semestre 3/Mantenimiento de softw...
.../adapters/in/ReportController.java            |  8 ++++
 2 files changed, 63 insertions(+), 1 deletion(-)

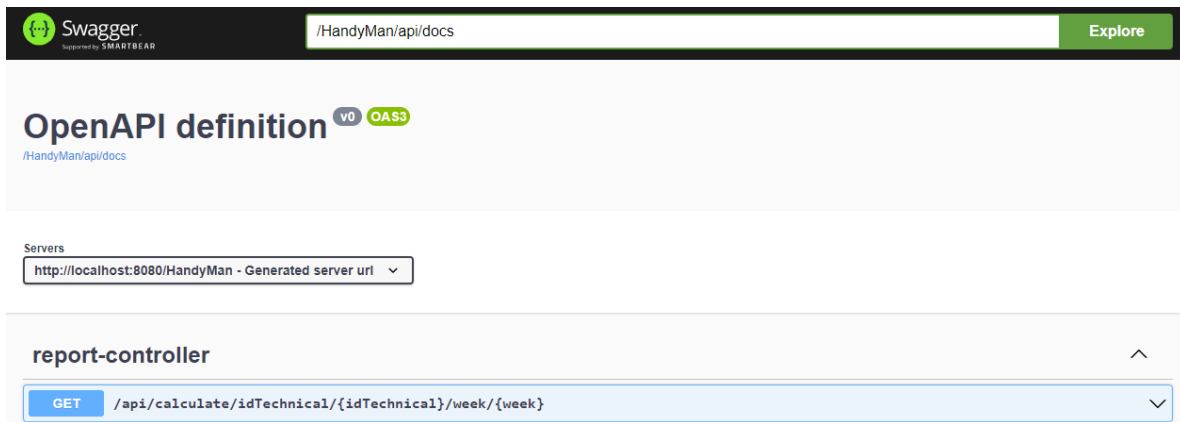
Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git push
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (16/16), 2.21 KiB | 753.00 KiB/s, done.
Total 16 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/BrayanBonilla-Ibero/HorasLaborales.git
 f20fd25..c43771c  main -> main

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$ git branch -d gettHour
Deleted branch gettHour (was c43771c).

Admin@LAPTOP-4992U9E2 MINGW64 ~/Documents/IBERO/Semestre 3/Mantenimiento de soft
ware/Unidad 2/Actividad 4/Git/HorasLaborales (main)
$
```



El resultado obtenido es el siguiente



#### 4.5 Historia de usuario 5: vista consulta horas

Se requiere diseñar la vista al usuario que le permita consultar las horas laboradas en una semana determinada.

Para llevar a cabo esta tarea vamos a crear una rama específica para poder trabajar esta hu, con el comando `git checkout -b apiDoc`

```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE
06:22:49  ■ HorasLaborales  ?main ?1 ~1  8 4ms
● → git checkout -b apiDoc
Switched to a new branch 'apiDoc'
06:26:05  ■ HorasLaborales  ?apiDoc ?1 ~1  8 149ms
○ →
```

Una vez trabaja la HU en el código fuente, procedemos a verificar los cambios, con el comando `git status`

```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

06:27:43  ■ HorasLaborales  ?apiDoc  ?1 ~3  8ms
• → git status
On branch apiDoc
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:   Actividad.docx
        modified:   handyman-fe/src/app/views/hour/hour.component.html
        modified:   handyman-fe/src/app/views/hour/hour.component.ts

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ~WRL0004.tmp

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

06:27:53  ■ HorasLaborales  ?apiDoc  ?1 ~3  170ms
```

Agregamos los cambios al stage, con el comando git add .

```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

06:28:09  ■ HorasLaborales  ?apiDoc  ?1 ~3  3ms
• → git add .
```

Realizamos el commit con los cambios realizados con el siguiente comando

```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

06:28:42  ■ HorasLaborales  ?apiDoc  +1 ~3  5ms
• → git commit -m "Visual de calculadora de horas"
[apiDoc 6a8902b] Visual de calculadora de horas
4 files changed, 95 insertions(+)
create mode 100644 ~WRL0004.tmp
```

Una vez realizado el commit, procedemos a pasar a la rama principal con el siguiente comando

```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE

06:30:33  ■ HorasLaborales  ?apiDoc  9ms
• → 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)
```

Una vez verificada la solución a la historia de usuario, procedemos a realizar el merge con la rama principal, con el siguiente comando

```
06:30:57 ■ HorasLaborales ○ ?main 13ms
• → git merge apiDoc
Merge made by the 'ort' strategy.
 handyman-fe/src/app/views/hour/hour.component.html | 42 ++++++
 handyman-fe/src/app/views/hour/hour.component.ts   | 53 ++++++
 2 files changed, 95 insertions(+)
06:31:04 ■ HorasLaborales ○ ?main 290ms
○ →
```

Subimos los cambios incorporados a la rama principal al repositorio remoto, con el siguiente comando

```
06:31:24 ■ HorasLaborales ○ ?main 4ms
• → git push
Enumerating objects: 22, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 4 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (13/13), 324.49 KiB | 3.09 MiB/s, done.
Total 13 (delta 9), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (9/9), completed with 6 local objects.
To https://github.com/BrayanBonilla-Ibero/HorasLaborales.git
 409ac23..a7236b7  main -> main
```

Como ya nos vamos a utilizar más la rama creada para trabajar la HU, procedemos a borrarla con el siguiente comando

```
06:32:01 ■ HorasLaborales ○ ?main 4ms
• → git branch -d apiDoc
Deleted branch apiDoc (was 6a8902b).
06:32:05 ■ HorasLaborales ○ ?main 157ms
○ →
```

El resultado obtenido es el siguiente

# Calculadora de horas

Identificación Técnico

Técnico

Número de la semana

Semana

CONSULTAR

Horas normales

Horas normales extras

Horas nocturnas

Horas nocturnas extras

Horas dominicales

Horas dominicales extras

< REGISTRO REPORTE DE SERVICIO

## 5 Documentación de comandos GIT usados

GIT es un software de control de versiones muy potente y ampliamente utilizado en la actualidad. De acuerdo con Sánchez Alberca (n.d.) el control de versiones hace referencia a la administración de los diferentes cambios que se realizan sobre los componentes de un sistema de software.

GIT es tan solo uno de los tantos software CVS existentes en la actualidad, según Gómez Bachiller (2015) existen otras alternativas como: Subversion, SourceSafe, ClearCase, Bazaar, Mercurial, etc.

### 5.1 git clone

Este es uno de los primeros comandos que empleamos cuando empezamos a trabajar con GIT en proyectos colaborativos, pues nos permite tener una copia en local de un repositorio que se encuentra de manera remota en GitHub. Siempre que se emplea git clone se coloca, luego del comando, la URL del proyecto que siempre termina en .git, como puede verse en el código del proyecto.

### 5.2 git status

Permite ver el estado actual del proyecto, mostrando los archivos que han sido modificados o agregados al stage área.

### 5.3 git add .

Este comando agrega los cambios que hemos realizado en uno o varios archivos al staging área, en este lugar se encuentran los datos del proyecto y los cambios que se han realizado sobre estos. Para entender mejor este concepto es importante repasar los tres estados del trabajo en GIT.

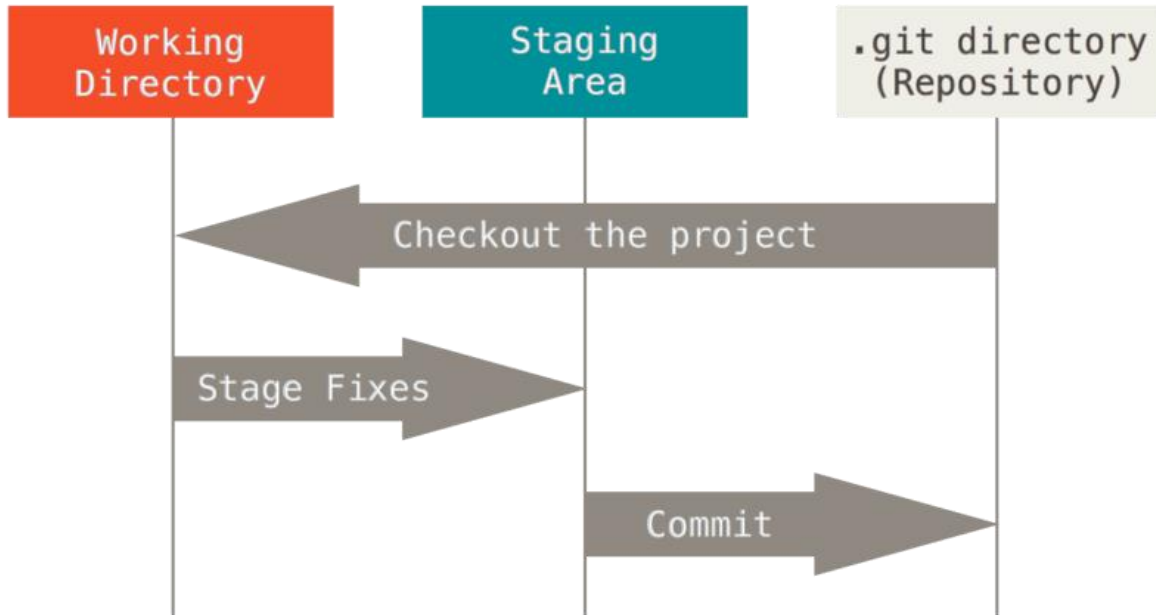


Figura 1. Directorio de trabajo, área de almacenamiento y el directorio Git. Fuente: Documentación oficial de GIT, recuperado de: <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Fundamentos-de-Git>

#### 5.4 git commit -m

El comando GIT commit permite realizar una captura de los cambios que han sido prepagrados y agregados con el comando git add, estas se consideran como “versiones seguras de un proyecto” (Atlassian, n.d.)

#### 5.5 git pull

El comando git pull nos permite traer toda la información desde un proyecto remoto hacia la copia del proyecto que tenemos en local, cuando estamos trabajando en equipo, por ejemplo, podemos hacer un git pull para que nuestro proyecto en local se actualice con los últimos cambios que pueda tener el proyecto en remoto.

#### 5.6 git push

Este comando realiza el trabajo contrario al comando anterior, pues en esta ocasión push nos permite enviar al remoto todos los cambios que tenemos efectuados en el repositorio local.

## **5.7 git checkout**

Este comando nos permite movernos entre ramas, si deseamos, por ejemplo, crear una nueva rama y ubicarnos en ella con un solo comando le podemos agregar el argumento `-b`, así lo empleamos en este proyecto.

## **5.8 git merge**

Nos permite fusionar ramas, esta es la forma que usa git para volver a unir un historial de trabajo que había sido bifurcado, cuando se crearon nuevas ramas anteriormente para desarrollar funcionalidades específicas, cuando estas funcionalidades han sido probadas y revisadas se pueden sumar a la rama principal a través de merge.

## **5.9 git branch**

El trabajo con ramas es una de las grandes posibilidades que brinda GIT, si bien este no es exclusivo de GIT, pues cualquier sistema de control de versiones actual presenta algún mecanismo para soportar el trabajo con ramas (Chacon & Straub, 2021, p. 58), su uso en GIT es bastante sencillo. Las ramas generalmente se usan para tener, en ciertos momentos del desarrollo, una copia del código del proyecto “aislado del resto del proyecto” para agregar nuevas funcionalidades, las cuales serán integradas posteriormente de nuevo a la rama principal.

## **5.10 git log**

Nos permite visualizar un historial de todos los commit realizados sobre un proyecto en particular. Podemos ver, entonces, cada uno de los commit (con su id único), el autor, la fecha y el mensaje que agregamos.

## 6 Referencias

Atlassian. (n.d.). *Aprende Git con Bitbucket cloud*.

<https://www.atlassian.com/es/git/tutorials/saving-changes/git-commit>

Chacon, S., & Straub, B. (2021). *Pro Git* (Segunda Ed). Apress.

<file:///C:/Users/jcmon/Downloads/progit.pdf>

Gómez Bachiller, S. (2015). *Introducción a GIT y GITHUB*.

<https://www.uco.es/aulasoftwarelibre/wp-content/uploads/2015/11/git-cosfera-dia-1.pdf>

Sánchez Alberca, A. (n.d.). *Introducción a GIT: sistema de control de versiones*.

<https://aprendeconalf.es/docencia/git/manual/manual-git.pdf>