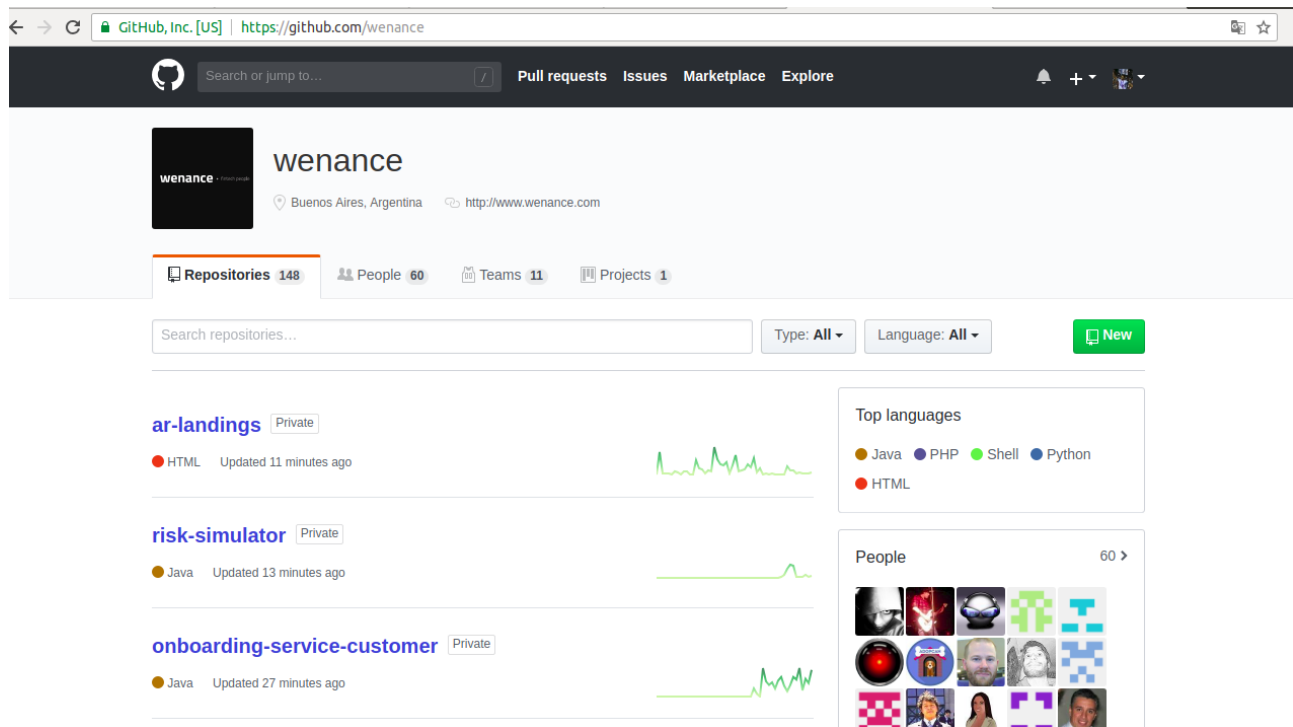


Capacitación GIT

Bajarse gitbash

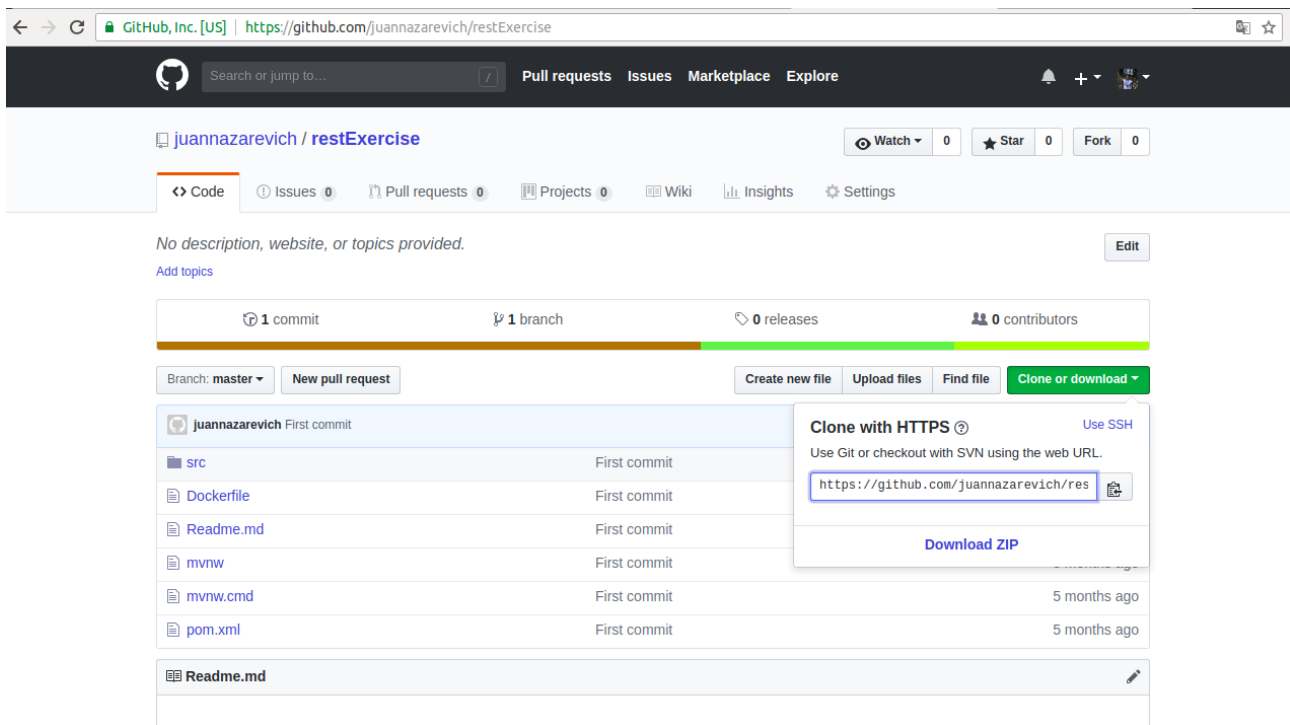
Ingresar a la página de github y elegir el proyecto

<https://github.com/wenance>



En el proyecto ir a “Clone or download” que nos provee la url para poder bajar el repo a nuestro local

<https://github.com/wenance/core-middleware.git>



En linea de comando :

git clone <https://github.com/wenance/core-middleware.git>

```
git clone https://github.com/juannazarevich/restExercise.git
```

Nos ubicamos dentro de la carpeta generada

```
juan@juan-ThinkPad-E470:~/Repos$ cd restExercise/  
juan@juan-ThinkPad-E470:~/Repos/restExercise$
```

Baja el repositorio actualizado con sus branches y se para en branch 'master'
Esto lo podemos chequear utilizando el comando:

git status

Si queremos movernos de branch, por ejemplo 'develop'

git checkout develop

```
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git checkout develop  
Cambiado a rama 'develop'  
juan@juan-ThinkPad-E470:~/Repos/restExercise$
```

Queremos encarar una nueva tarea, creamos un nuevo branch de forma local. Por convención usamos la siguiente nomenclatura 'feature/{titulo-de-la-tarea}'

git checkout -b feature/capacitacion-git

```
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git checkout -b feature/capa-git  
Cambiado a nueva rama 'feature/capa-git'
```

Realizamos algún cambio en algún archivo ó creamos un nuevo archivo. En ese caso se verá en rojo y todavía no está agregado al stage
En el caso de querer agregar todos los cambios generados ó lo nuevo que creamos

`git add .`

Sino: podemos agregar archivo por archivo

`git add {path del archivo}`

```
comando a nueva rama feature/capa-git
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git status
En la rama feature/capa-git
Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

    comandos-git.txt

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git add comandos-git.txt
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git status
En la rama feature/capa-git
Cambios a ser confirmados:
  (usa "git reset HEAD <archivo>..." para sacar del área de stage)

    nuevo archivo: comandos-git.txt
```

si queremos sacar el archivo de stage (sin seguimiento)

`git reset {path del archivo}`

```
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git reset comandos-git.txt
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git status
En la rama feature/capa-git
Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

    comandos-git.txt

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
```

Una vez en stage, debemos commitear los cambios y agregarle un mensaje. De esta forma se actualiza nuestra versión del repo y es la forma de asentar nuestros cambios previo a pushear los cambios.

`git commit -m "Agrego nuevo archivo con comandos de git"`

```
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git commit -m "Agrego nuevo archivo con comandos de git"
[feature/capa-git df16f85] Agrego nuevo archivo con comandos de git
1 file changed, 5 insertions(+)
create mode 100644 comandos-git.txt
```

Ahora podemos subir nuestros cambios al servidor. Cuando hagamos esto, además vamos a estar agregando nuestro branch local.

`git push origin feature/capacitacion-git`

```

Create Mode 100644 comandos-git.txt
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git status
En la rama feature/capa-git
nada para hacer commit, el árbol de trabajo esta limpio
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git push origin feature/capa-git
Username for 'https://github.com': juannazarevich
Password for 'https://juannazarevich@github.com':
Contando objetos: 3, listo.
Delta compression using up to 4 threads.
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 345 bytes | 345.00 KiB/s, listo.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/juannazarevich/restExercise.git
 * [new branch]      feature/capa-git -> feature/capa-git

```

Es muy común que en el repo haya cambios con bastante periodicidad, por lo que antes de mandar un PR y para evitar conflictos, vamos a mergear los cambios desde develop a nuestro branch feature/capa-git

Primero nos movemos utilizando el comando git checkout a la rama develop. Posteriormente ejecutamos un pull, con el cual vamos a traernos los cambios desde esta rama. (en mi caso figura actualizado porque efectué los cambios desde mi usuario)

```

juan@juan-ThinkPad-E470:~/Repos/restExercise$ git checkout develop
Cambiado a rama 'develop'
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git pull origin develop
Desde https://github.com/juannazarevich/restExercise
 * branch      develop    -> FETCH_HEAD
Ya está actualizado.

```

Luego nos vemos a nuestra rama de trabajo feature/capa-git y ejecutamos el comando git merge develop

En nuestro caso se abrirá una pantalla con un editor de texto (Nano) para poder editar el mensaje de commit que se registrará cuando hagamos el merge. Por default, se autogenera un mensaje descriptivo de la acción, por lo que podemos dejarlo así.

Para salir del pantalla ejecutamos ctrl + X y luego : q ENTER

```
GNU nano 2.9.3 /home/juan/Repos/restExercise/.git/MERGE_MSG
Merge branch 'develop' into feature/capa-git

# Por favor ingrese un mensaje de commit que explique por qué es necesaria esta fusión,
# especialmente si esto fusiona un upstream actualizado en una rama de tópico.
#
# Líneas comenzando con '#' serán ignoradas, y un mensaje vacío aborta
# el commit.

[ 7 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Texto ^J Justificar ^C Posición M-U Deshacer M-A Marcar text M-J A llave
^X Salir ^R Leer fich. ^A Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea M-E Rehacer M-G Copiar txt M-W Siguiente
```

De esta forma nos indica que se asentaron los cambios del branch develop en nuestro branch local

```
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git checkout feature/capa-git
Cambiado a rama 'feature/capa-git'
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git status
En la rama feature/capa-git
nada para hacer commit, el árbol de trabajo esta limpio
juan@juan-ThinkPad-E470:~/Repos/restExercise$ git merge develop
Merge made by the 'recursive' strategy.
 nuevo-archivo.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 nuevo-archivo.txt
```