



UAEM

Universidad Autónoma
del Estado de México



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO ZUMPANGO
INGENIERÍA EN COMPUTACIÓN

EXAMEN PRIMER PARCIAL: EVIDENCIAS GITHUB

UNIDAD DE APRENDIZAJE:

DISEÑO DE SISTEMAS

DOCENTE:

HAZEM ÁLVAREZ RODRÍGUEZ

ALUMNO:

LUIS GUSTAVO ROJAS PAREDES




1. Desde la cuenta personal de GitHub se creó un repositorio que tiene por nombre **Examen1erPar**, será un repositorio público.

Create a new repository


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


Owner * Repository name *

 luisgrp001 / Examen1erPar ✓

Great repository names are short and memorable. Need inspiration? How about **solid-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**

2. Dentro de la carpeta donde se encuentran los archivos de código Java generados y desarrollados se inicializará un nuevo proyecto de Git con el comando **git init**, posteriormente se crea un nuevo Branch llamado **Programa**, y finalmente se agregan todos los archivos de código al proyecto con **git add**.

```
MINGW64/c/Users/USUARIO/Desktop/examen
$ git add .
warning: in the working copy of 'Banco.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Contacto.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Sesion.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Usuario.java', LF will be replaced by CRLF the next time Git touches it
USUARIO@CUZ-36 MINGW64 ~/Desktop/examen (Programa)
```

3. Se realiza el commit correspondiente a la carga de los archivos de código con el comando **git commit -m "Archivos JAVA Rappi"**.

```
USUARIO@CUZ-36 MINGW64 ~/Desktop/examen (Programa)
$ git commit -m "Archivos JAVA Rappi"
[Programa (root-commit) ce9b024] Archivos JAVA Rappi
4 files changed, 233 insertions(+)
create mode 100644 Banco.java
create mode 100644 Contacto.java
create mode 100644 Sesion.java
create mode 100644 Usuario.java
```

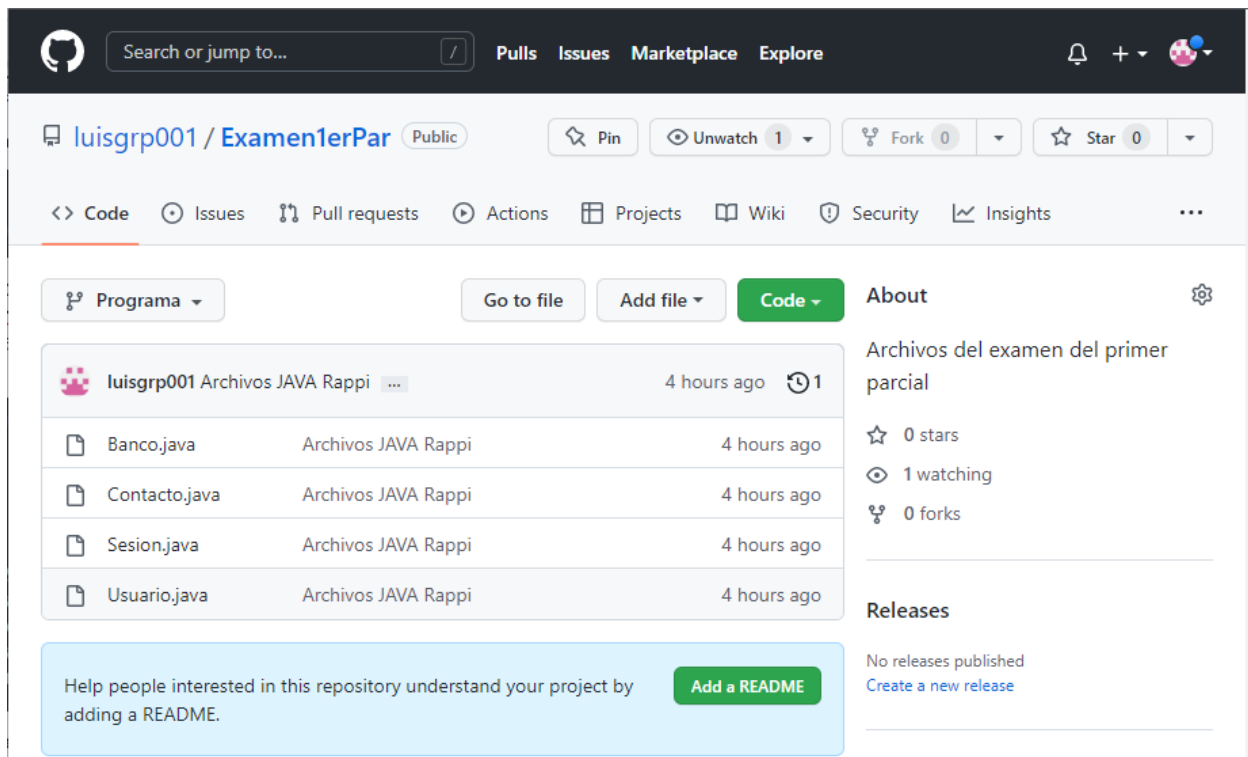
4. Con el proyecto Git actualizado localmente, podremos establecer la conexión entre el repositorio local y el repositorio remoto **Examen1erPar** utilizando **git remote add origin <https://github.com/luisgrp001/Examen1erPar.git>**. Si la conexión se establece correctamente al repositorio remoto, el comando sólo se ejecuta sin mostrar mensaje alguno en la terminal.

```
USUARIO@CUZ-36 MINGW64 ~/Desktop/examen (Programa)
$ git remote add origin https://github.com/luisgrp001/Examen1erPar.git
```

5. Tomaremos los archivos del repositorio local y los enviaremos hacia el repositorio remoto con el comando **git push -u origin Programa**, con lo que le indicamos a Git que cargará los archivos de código Java en el Branch **Programa** del repositorio **Examen1erPar**.

```
USUARIO@CUZ-36 MINGW64 ~/Desktop/examen (Programa)
$ git push -u origin Programa
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.78 KiB | 911.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/luisgrp001/Examen1erPar.git
* [new branch]      Programa -> Programa
branch 'Programa' set up to track 'origin/Programa'.
```

6. Al terminar de cargar los archivos se puede comprobar que los 4 archivos de código han sido agregados al repositorio remoto accediendo a éste en la cuenta personal de GitHub:



- Se crea un nuevo Branch llamado **Diagrama**, el cual contendrá todos los diagramas realizados para el proceso de Envío de dinero a través de la app de RappiPay. El comando utilizado es **git checkout -b Diagrama**.

```
USUARIO@CUZ-36 MINGW64 ~/Desktop/diagrama (master)
$ git checkout -b Diagrama
Switched to a new branch 'Diagrama'
```

- Desde la carpeta donde se encuentran los diagramas, se agregan todos los archivos al proyecto de Git local con el comando **git add .** y luego se realiza el snapshot que le corresponde mediante **commit -m "Diagramas de Rappi"**.

```
USUARIO@CUZ-36 MINGW64 ~/Desktop/diagrama (Diagrama)
$ git add .

USUARIO@CUZ-36 MINGW64 ~/Desktop/diagrama (Diagrama)
$ git commit -m "Diagramas de Rappi"
[Diagrama (root-commit) 6c9c324] Diagramas de Rappi
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 ClasesRappi.pdf
create mode 100644 EstadosRappi.pdf
create mode 100644 rappiBPMN.jpg
```

- Se enviarán los últimos archivos cargados al repositorio local hacia el repositorio remoto en el Branch **Diagrama** con el comando **git push -u origin Diagrama**.

```

USUARIO@CUZ-36 MINGW64 ~/Desktop/diagrama (Diagrama)
$ git push -u origin Diagrama
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 186.72 KiB | 18.67 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'Diagrama' on GitHub by visiting:
remote:   https://github.com/luisgrp001/Examen1erPar/pull/new/Diagrama
remote:
To https://github.com/luisgrp001/Examen1erPar.git
 * [new branch]      Diagrama -> Diagrama
branch 'Diagrama' set up to track 'origin/Diagrama'.

USUARIO@CUZ-36 MINGW64 ~/Desktop/diagrama (Diagrama)

```

10. Se comprueba que los archivos agregados en el paso 7 han sido cargados al repositorio **Examen1erPar** en un nuevo Branch nombrado **Diagrama**, desde el sitio del repositorio remoto.

The screenshot shows the GitHub interface for the repository 'luisgrp001 / Examen1erPar'. The 'Diagrama' branch is selected, showing a commit that is 1 commit ahead and 1 commit behind the 'Programa' branch. The commit message is 'Diagramas de Rappi'. The commit details show three files: 'ClasesRappi.pdf', 'EstadosRappi.pdf', and 'rappiBPMN.jpg', all uploaded 5 hours ago. The repository has 0 stars, 1 watching, and 0 forks. There are no releases published. A blue banner at the bottom encourages adding a README.

11. A continuación, se muestra una prueba del funcionamiento del código desarrollado a partir de los archivos generados por el diagrama de clases de StarUML:

```
C:\> Símbolo del sistema

C:\Users\fampa\OneDrive\Escritorio\Evidencias\RappiPay>javac *.java

C:\Users\fampa\OneDrive\Escritorio\Evidencias\RappiPay>java Sesion
USTED ES: Luis Rojas Paredes
SU SALDO ES DE: 8000.0 MXN
INGRESE EL MONTO A ENVIAR:2000
SU SALDO AHORA ES DE: 6000.0
```

12. Este documento donde se presentaron todas las capturas con los pasos realizados para generar el repositorio con los archivos del examen será cargado a un nuevo Branch llamado **Evidencias**. Por lo que éste archivo debe estar situado en el repositorio de GitHub **Examen1erPar** en el Branch **Evidencias**.

```
fampa@DESKTOP-GAS4UI8 MINGW64 ~/OneDrive/Escritorio/Evidencias/RappiPay (master)
$ git checkout -b Evidencias
Switched to a new branch 'Evidencias'
```

LINK AL REPOSITORIO:

<https://github.com/luisgrp001/Examen1erPar.git>