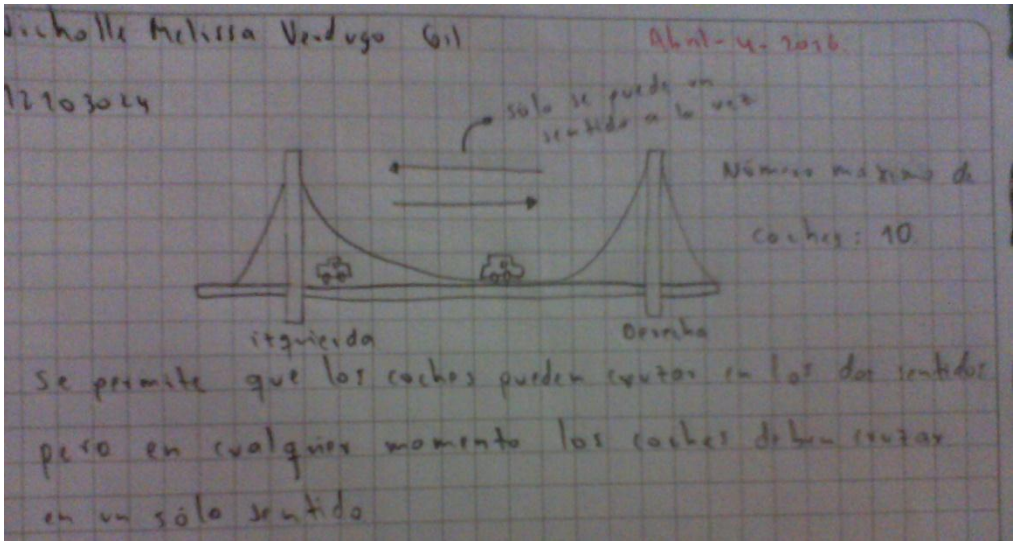
Nombre: Nicholle Melissa Verdugo Gil
Código: 12103024
Sistemas operacionales
URL del repositorio: https://github.com/NicholleVerdugo

PARCIAL 1

1. **Sincronización de procesos**
   a. **Gráfico que ilustra el problema**



En el gráfico se puede observar que hay un puente y un flujo de coches. Los coches se pueden mover en ambos sentidos pero en un momento dado sólo pueden cruzar el puente en un sentido ya sea de izquierda a derecha o de derecha a izquierda.

El puente permite una capacidad máxima de 10 coches.

No se permite que circulen coches en sentidos contrarios en un instante.

   b. **Estructuras de sincronización utilizadas**

Para realizar la sincronización de los procesos, se tiene en el programa una clase **Puente**, la cual contiene los siguientes métodos:
InicioSentidoDchaAIzqda(),FinSentidoDchaAIzqda(),
InicioSentidoIzqdaADcha (), FinSentidoIzqdaADcha ()
Éstos deben proveer exclusión mutua, para evitar el ingreso al contador de coches por más de un proceso a la vez, ya que es el recurso compartido.

Hay un sólo recurso que es el Puente.
Si un carro de la derecha hace InicioSentidoDchaAIzqda() debe solicitar el recurso.
Esto sólo lo hace el primer carro de la derecha, que es el que obtiene el recurso y los demás ya pueden cruzar normalmente.
También, si el carro de la izquierda quiere cruzar el puente, debe esperar hasta que se libere el recurso.

Para realizar el control del cruce del puente se va a utilizar un semáforo, se hace P() en el semáforo de Puente cuando un coche de la izquierda quiere cruzar, será concedido solamente si el semáforo es 1, esto es, no hay coches en el lado derecho. Los coches de la derecha harán V() para liberar el puente solamente cuando hayan salido todos, es decir, numCoches es 0. Cuando un coche de la derecha ingresa al puente, es decir, que hace P() y se le concede el recurso, no se retornará V() sino hasta que salgan todos, que es con la variable maxCoches, cuando ésta es liberada.

Las variables utilizadas para la sincronización son las siguientes:

- **int numCoches:** Es la variable que representa el contador de los coches que van cruzando el puente, para tener el control de cuántos coches pasan de derecha a izquierda y cuántos de izquierda a derecha. Ésta variable es el recurso compartido.

- **Semaphore mutexNumCoches:** Esta variable es el semáforo que regula la variable numCoches, es decir, que permite la adquisición y la liberación del recurso. Garantiza que sólo un proceso tenga acceso a la vez.

```
public void InicioSentidoDchaAIzqda() throws InterruptedException {
    sentido.acquire();//Si ya los coches de la derecha lo están usanc
    maxCoches.acquire();//Se adquiere la capacidad máxima de coches c
    mutexNumCoches.acquire();
    numCoches ++;
    if (numCoches == 1) {
        puente.acquire();
    }
    mutexNumCoches.release();//Debe ir adentro el acquire sino, algui
    sentido.release();
}

public void FinSentidoDchaAIzqda() throws InterruptedException {
    mutexNumCoches.acquire();
    numCoches --;
    mutexNumCoches.release();
    if (numCoches == 0) { // Si esta adentro de mutex se da mas impor
        puente.release();
        maxCoches.release();
    }

}
```

- **Semaphore maxCoches:** Esta variable es el semáforo que representa el peso máximo que soporta el puente, en este caso el número máximo son 10 coches simultáneamente. Se realiza P() en los métodos de inicio para adquirir el número de coches que pueden cruzar el puente, simultáneamente y se realiza V() en los métodos de fin para liberar el número de coches que pueden cruzar el puente. Lo anterior se da para los dos sentidos.

```
public void InicioSentidoDchaAIzqda() throws InterruptedException {
    sentido.acquire();//Si ya los coches de la derecha lo están usan
    maxCoches.acquire();//Se adquiere la capacidad máxima de coches
    mutexNumCoches.acquire();
    numCoches ++;
    if (numCoches == 1) {
        puente.acquire();
    }
    mutexNumCoches.release();//Debe ir adentro el acquire sino, algu
    sentido.release();
}

public void FinSentidoDchaAIzqda() throws InterruptedException {
    mutexNumCoches.acquire();
    numCoches --;
    mutexNumCoches.release();
    if (numCoches == 0) { // Si esta adentro de mutex se da mas impo
        puente.release();
        maxCoches.release();
    }

}
```

- **Semaphore puente:** Esta variable es el semáforo que representa el puente y dependiendo del sentido que se tome ya sea de derecha a izquierda o de izquierda a derecha, se realiza V() en los métodos de fin para liberar el recurso cuando numCoches es 0 y se realiza P() en los métodos de inicio para adquirir el recurso, que son los coches que van a cruzar el puente, cuando numCoches es 1.

```
public void InicioSentidoDchaAIzqda() throws InterruptedException {
    sentido.acquire();//Si ya los coches de la derecha lo están usar
    maxCoches.acquire();//Se adquiere la capacidad máxima de coches
    mutexNumCoches.acquire();
    numCoches ++;
    if (numCoches == 1) {
        puente.acquire();
    }
    mutexNumCoches.release();//Debe ir adentro el acquire sino, algu
    sentido.release();
}

public void FinSentidoDchaAIzqda() throws InterruptedException {
    mutexNumCoches.acquire();
    numCoches --;
    mutexNumCoches.release();
    if (numCoches == 0) { // Si esta adentro de mutex se da mas impc
        puente.release();
        maxCoches.release();
    }

}
```

- **Semaphore sentido:** Esta variable es el semáforo que representa el sentido. En un momento dado sólo se permite a los coches cruzar en un mismo sentido de circulación. Si ya los coches de la derecha lo están usando no puede aumentar los Coches de la izquierda y viceversa. Se realiza P() sólo al principio de los métodos de inicio para adquirir el recurso, que en este caso es el sentido que van a tomar los coches para cruzar el puente y se realiza V() sólo al final de los métodos de inicio para liberar el recurso y el otro sentido lo pueda adquirir.

```
public void InicioSentidoDchaAIzqda() throws InterruptedException {
    sentido.acquire();//Si ya los coches de la derecha lo están usar
    maxCoches.acquire();//Se adquiere la capacidad máxima de coches
    mutexNumCoches.acquire();
    numCoches ++;
    if (numCoches == 1) {
        puente.acquire();
    }
    mutexNumCoches.release();//Debe ir adentro el acquire sino, algu
    sentido.release();
}
```

Debe existir una sincronización entre procesos de tal manera que cuando se esté modificando un registro, se impida el acceso al dato. Esta sincronización se conoce como exclusión mutua.

También, se evita la inanición porque se permite que pasen todos los recursos de un sentido primero sin que el puente se sobrecargue y luego se le da el paso a los coches del otro sentido para que crucen el puente.

c. El código se encuentra en la carpeta Puente-NicholleVerdugo.

d. Caso que representa la sincronización de los procesos

La clase **Test** es donde se realizan las pruebas de la sincronización.

CochesDcha.java | CochesIzqda.java | Puente.java | Test.java

```
1  package co.edu.icesi.puente;
2
3  public class Test {
4
5      public static void main(String args[]) {
6
7          Puente p = new Puente();
8
9          for (int i = 1; i < 11; i++) {
10             CochesIzqda p1 = new CochesIzqda(p,i);
11             p1.start();
12         }
13
14         for (int i = 1; i < 11; i++) {
15             CochesDcha p2 = new CochesDcha(p, i);
16             p2.start();
17         }
18     }
19 }
20
```

Markers | Properties | Servers | Data Source Explorer | Snippets | Console

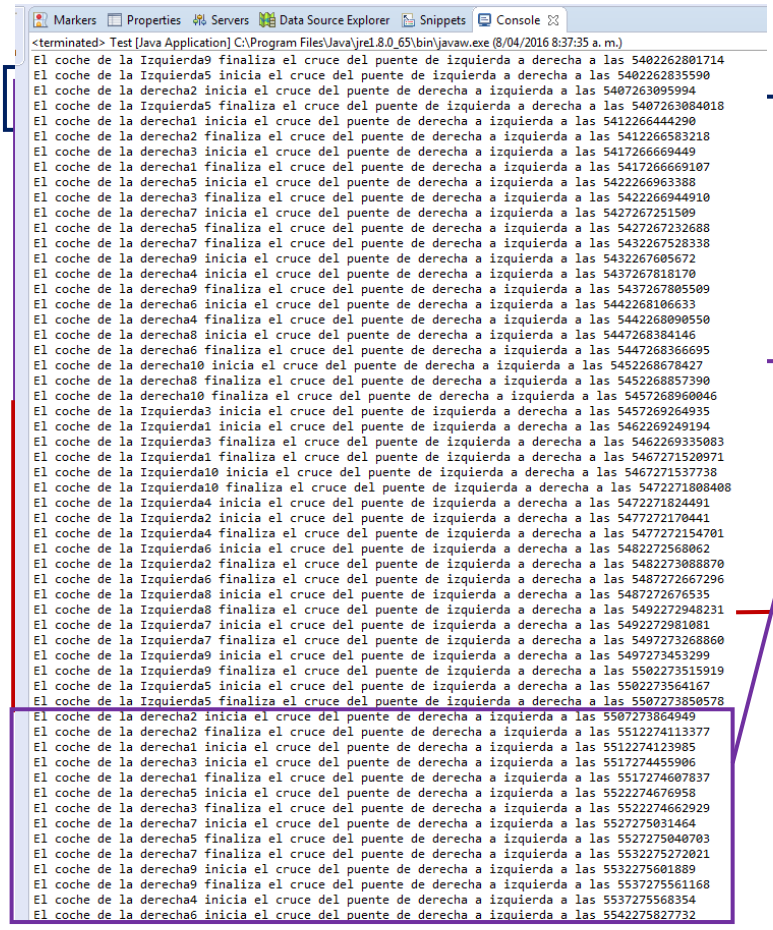
<terminated> Test [Java Application] C:\Program Files\Java\jre1.8.0_65\bin\javaw.exe (8/04/2016 8:37:35 a. m.)

```
El coche de la Izquierda1 inicia el cruce del puente de izquierda a derecha a las 5302254610793
El coche de la Izquierda3 inicia el cruce del puente de izquierda a derecha a las 5302254893439
El coche de la Izquierda10 inicia el cruce del puente de izquierda a derecha a las 5302254956402
El coche de la Izquierda8 inicia el cruce del puente de izquierda a derecha a las 5302256277584
El coche de la Izquierda6 inicia el cruce del puente de izquierda a derecha a las 5302256351496
El coche de la Izquierda4 inicia el cruce del puente de izquierda a derecha a las 5302256477763
El coche de la Izquierda2 inicia el cruce del puente de izquierda a derecha a las 5302256535593
El coche de la Izquierda9 inicia el cruce del puente de izquierda a derecha a las 5302257029026
El coche de la Izquierda7 inicia el cruce del puente de izquierda a derecha a las 5302257088566
El coche de la Izquierda5 inicia el cruce del puente de izquierda a derecha a las 5302257136815
El coche de la Izquierda3 finaliza el cruce del puente de izquierda a derecha a las 5307254393847
El coche de la Izquierda1 finaliza el cruce del puente de izquierda a derecha a las 5307254394189
El coche de la Izquierda10 finaliza el cruce del puente de izquierda a derecha a las 5307255370789
El coche de la Izquierda4 finaliza el cruce del puente de izquierda a derecha a las 5307256326175
El coche de la Izquierda2 finaliza el cruce del puente de izquierda a derecha a las 5307256312145
El coche de la Izquierda6 finaliza el cruce del puente de izquierda a derecha a las 5307256609163
El coche de la Izquierda8 finaliza el cruce del puente de izquierda a derecha a las 5307256617033
El coche de la Izquierda9 finaliza el cruce del puente de izquierda a derecha a las 5307257370870
El coche de la Izquierda7 finaliza el cruce del puente de izquierda a derecha a las 5307257379425
El coche de la Izquierda5 finaliza el cruce del puente de izquierda a derecha a las 5307257518353
El coche de la derecha2 inicia el cruce del puente de derecha a izquierda a las 5307257587474
El coche de la derecha1 inicia el cruce del puente de derecha a izquierda a las 5312257691499
El coche de la derecha2 finaliza el cruce del puente de derecha a izquierda a las 5312257858144
El coche de la derecha1 finaliza el cruce del puente de derecha a izquierda a las 5317257935136
El coche de la derecha3 inicia el cruce del puente de derecha a izquierda a las 5317257951903
El coche de la derecha3 finaliza el cruce del puente de derecha a izquierda a las 5322258244815
El coche de la derecha5 inicia el cruce del puente de derecha a izquierda a las 5322258487083
El coche de la derecha5 finaliza el cruce del puente de derecha a izquierda a las 5327258513431
El coche de la derecha7 inicia el cruce del puente de derecha a izquierda a las 5327258531225
El coche de la derecha7 finaliza el cruce del puente de derecha a izquierda a las 5332258800184
El coche de la derecha9 inicia el cruce del puente de derecha a izquierda a las 5332258827559
El coche de la derecha9 finaliza el cruce del puente de derecha a izquierda a las 5337259080092
El coche de la derecha4 inicia el cruce del puente de derecha a izquierda a las 5337259217309
El coche de la derecha6 inicia el cruce del puente de derecha a izquierda a las 5342259376426
El coche de la derecha4 finaliza el cruce del puente de derecha a izquierda a las 5342259374715
El coche de la derecha8 inicia el cruce del puente de derecha a izquierda a las 5347259673102
El coche de la derecha6 finaliza el cruce del puente de derecha a izquierda a las 5347259654282
El coche de la derecha8 finaliza el cruce del puente de derecha a izquierda a las 5352260027949
El coche de la derecha10 inicia el cruce del puente de derecha a izquierda a las 5352260048138
El coche de la Izquierda3 inicia el cruce del puente de izquierda a derecha a las 5357260250713
El coche de la derecha10 finaliza el cruce del puente de derecha a izquierda a las 5357260230866
El coche de la Izquierda1 inicia el cruce del puente de izquierda a derecha a las 5362260524120
El coche de la Izquierda3 finaliza el cruce del puente de izquierda a derecha a las 5362260510090
El coche de la Izquierda1 finaliza el cruce del puente de izquierda a derecha a las 5367260808819
El coche de la Izquierda10 inicia el cruce del puente de izquierda a derecha a las 5367260816689
El coche de la Izquierda4 inicia el cruce del puente de izquierda a derecha a las 5372261096940
El coche de la Izquierda10 finaliza el cruce del puente de izquierda a derecha a las 5372261082226
El coche de la Izquierda2 inicia el cruce del puente de izquierda a derecha a las 5377261375480
El coche de la Izquierda4 finaliza el cruce del puente de izquierda a derecha a las 5377261365215
El coche de la Izquierda6 inicia el cruce del puente de izquierda a derecha a las 5382261692003
El coche de la Izquierda2 finaliza el cruce del puente de izquierda a derecha a las 5382262045139
El coche de la Izquierda6 finaliza el cruce del puente de izquierda a derecha a las 5387261961646
El coche de la Izquierda8 inicia el cruce del puente de izquierda a derecha a las 5387261982861
El coche de la Izquierda8 finaliza el cruce del puente de izquierda a derecha a las 5392262257637
El coche de la Izquierda7 inicia el cruce del puente de izquierda a derecha a las 5392262651152
El coche de la Izquierda7 finaliza el cruce del puente de izquierda a derecha a las 5397262544390
El coche de la Izquierda9 inicia el cruce del puente de izquierda a derecha a las 5397262550891
```

Representa la ejecución de los coches de la izquierda, entonces se puede observar que inician y terminan en un tiempo determinado y que no excede la capacidad máxima del puente.

Representa la ejecución de los coches de la derecha, entonces se puede observar que inician y terminan en un tiempo determinado y que no excede la capacidad máxima

Aquí se puede ver que pareciera que se mezclaran los sentidos, pero esto ocurre porque siempre se demora en la escritura de la línea de salida, pero si nos fijamos en el tiempo nos damos cuenta que los sentidos no se contraponen



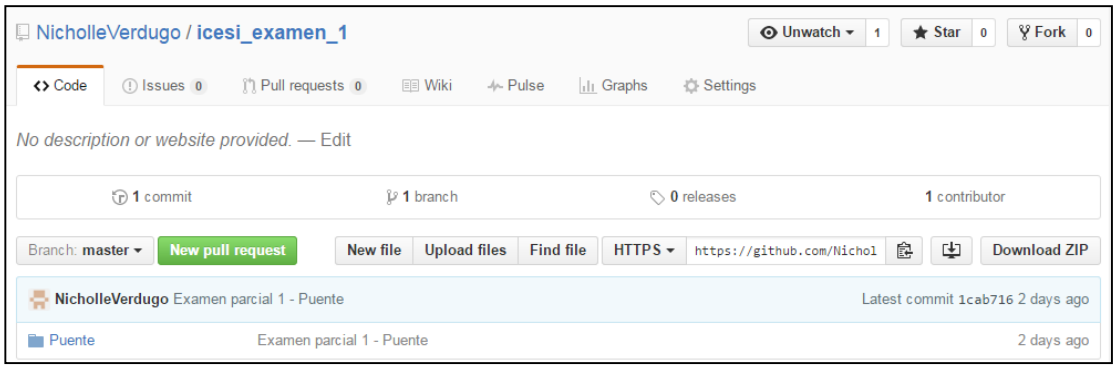
Este es otro caso en donde la escritura de la línea de salida se demora pero no aun así no se presenta sobrecarga en el puente. Se respeta en número de carros máximo y el sentido.

Representa la ejecución de los coches de la derecha, entonces se puede observar que inician y terminan en un tiempo determinado y que no excede la capacidad máxima

Representa la ejecución de los coches de la izquierda, entonces se puede observar que inician y terminan en un tiempo determinado y que no excede la capacidad máxima del puente. En cuanto a los hilos de los sentidos, primero se ejecuta en un sentido y luego en el otro respetando la capacidad y el sentido.

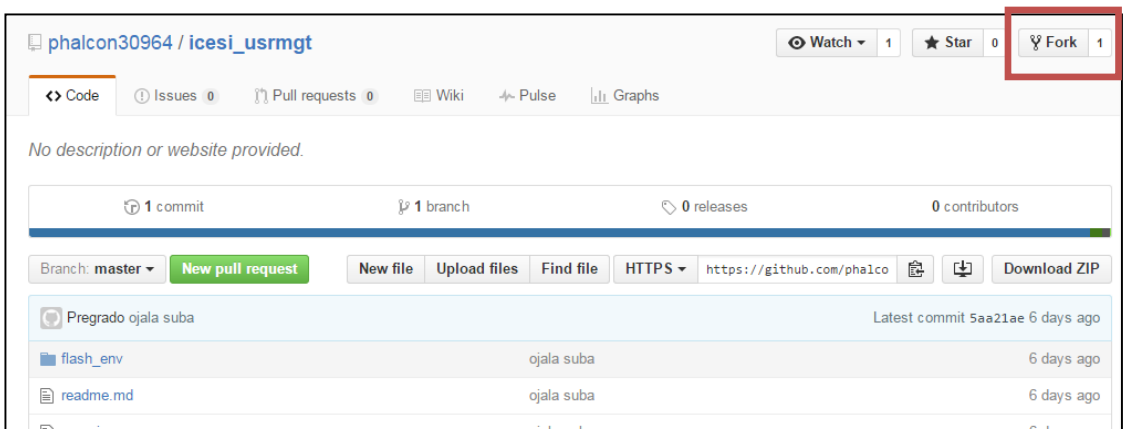
2. Git

Para el almacenamiento de las fuentes del examen se creó un repositorio con el nombre Icesi_examen_1

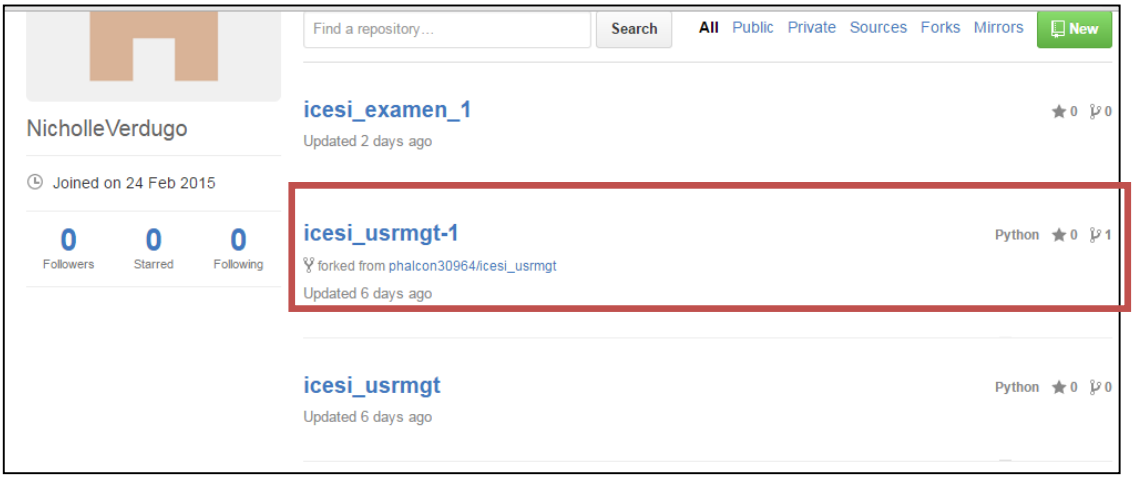


3. Fork

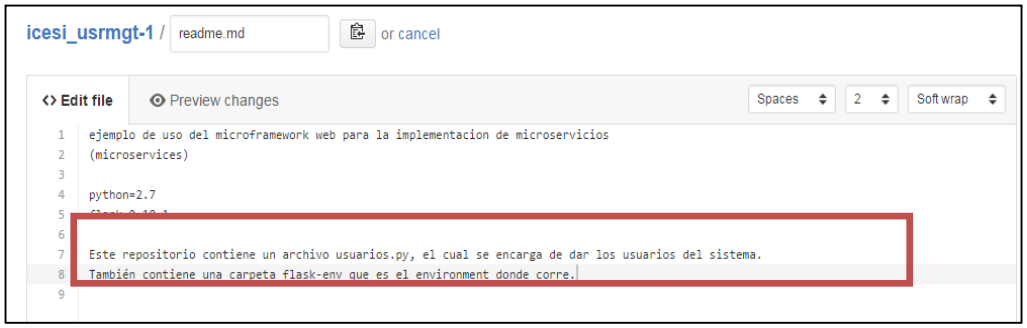
Aquí realizo un fork al repositorio phalcon30964/Icesi_usrmgt



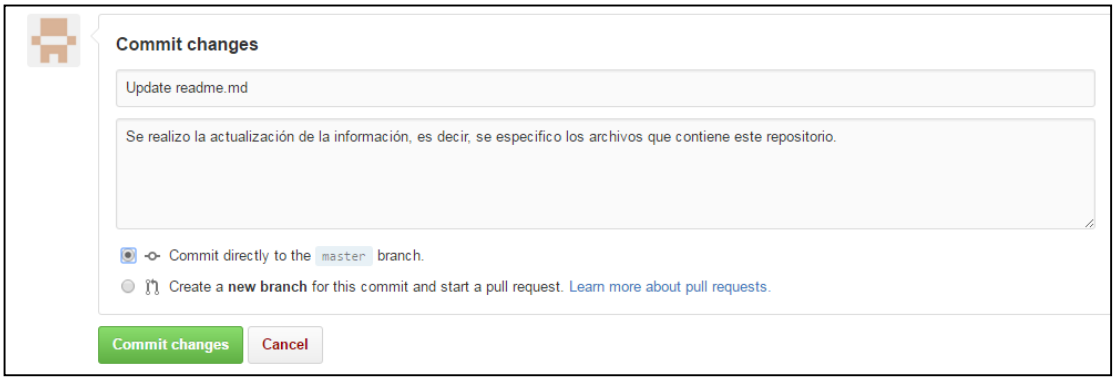
Ya se puede evidenciar que he realizado el fork, porque tengo una copia en mis repositorios.



Se realizó una modificación en el readme.md

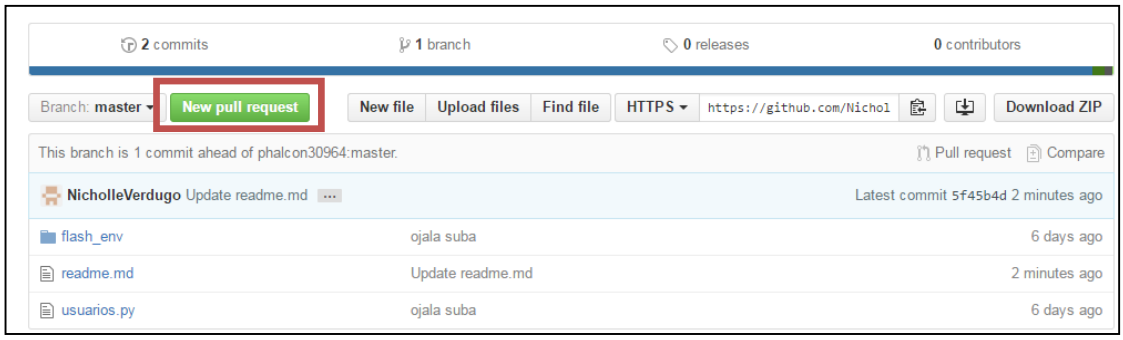


Se colocó el comentario pertinente a la actualización

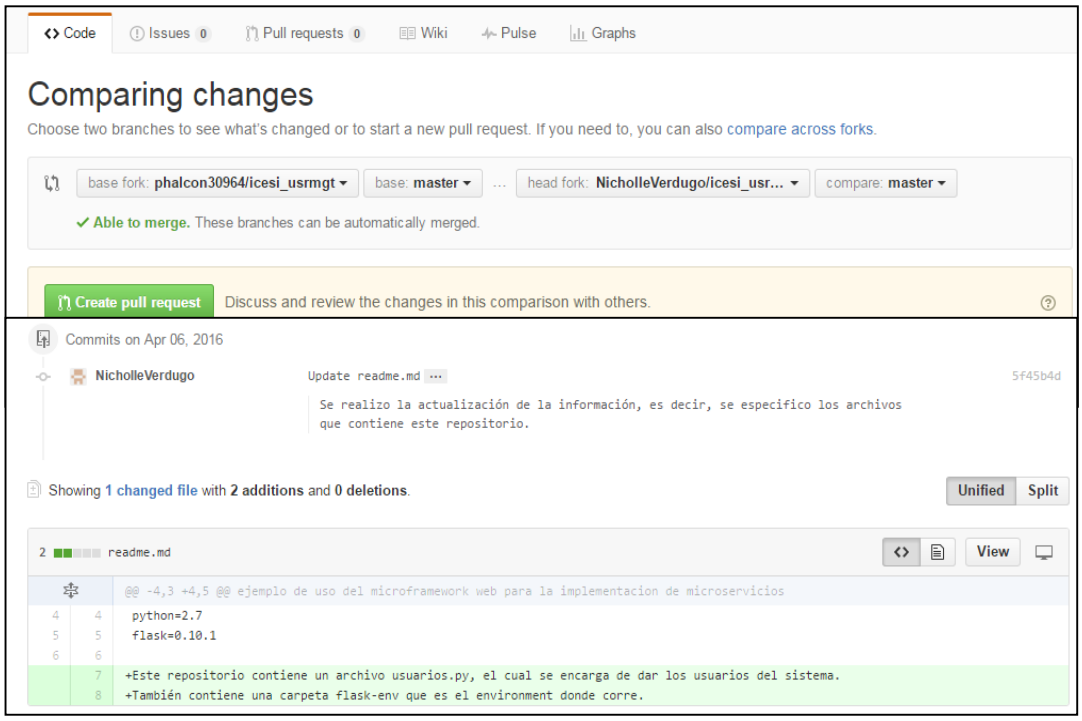


Aparece que se realizó una contribución al repositorio

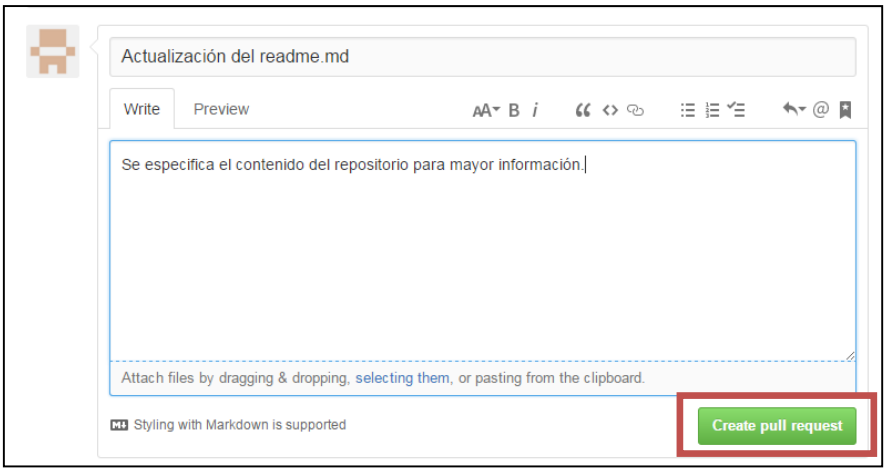
Luego realizó un pull request con los cambios realizados



Aquí se evidencia que la contribución fue exitosa



Y entonces ya puedo crear el pull request con su respectivo comentario de la actualización del readme.md



Luego se realiza el pull request y obtengo que se ha realizado la actualización.