	Carátula para entrega de prácticas	
Facultad de Ingeniería		Laboratorio de docencia

Laboratorios de computación

Salas A y B

<i>Profesor:</i>	Ing. José Antonio Ayala Barbosa
<i>Asignatura:</i>	Programación Orientada a Objetos
<i>Grupo:</i>	1
<i>No de Práctica(s):</i>	Práctica 6 Organización de Clases
<i>Integrante(s):</i>	José Luis Arroyo Chavarría Francisco Moisés Barrera Guardia
<i>No. de Equipo de cómputo empleado:</i>	
<i>No. de Lista o Brigada:</i>	
<i>Semestre:</i>	3
<i>Fecha de entrega:</i>	12/11/2020
<i>Observaciones:</i>	

CALIFICACIÓN: _____

I. Previo

1. ¿Qué es git?

Git es un sistema de control de versiones. Nos sirve para trabajar en equipo de una manera mucho más simple y optima cuando se está desarrollando un software.

Cuando acabamos de desarrollar nuestro código, utilizamos Git para mezclar los cambios con los otros compañeros

2. ¿Qué es github?

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

3. ¿Qué es el tablero canvas de github?

Contienen cambios de código destinados a corregir errores, mejorar el rendimiento y prepararse para nuevas funciones. No afectan los flujos de trabajo de los clientes.

4. ¿Cuáles son los comandos básicos de github?

- **git help**

Muestra una lista con los comandos más utilizados en GIT.

- **git init**

Crea localmente un repositorio con GIT y así utilizar todo el funcionamiento que GIT ofrece. Basta con estar ubicados dentro de la carpeta donde tenemos nuestro proyecto y ejecutar el comando. Cuando agreguemos archivos y un commit, se va a crear el branch master por defecto.

- **git add + path**

Agrega al repositorio los archivos que indiquemos.

- **git add -A**

Agrega al repositorio todos los archivos y carpetas que estén en nuestro proyecto, los cuales GIT no está siguiendo.

- **git commit -m "mensaje" + archivos**

Hace commit a los archivos que indiquemos, de esta manera quedan guardados nuestras modificaciones.

- **git commit -am "mensaje"**

Hace commit de los archivos que han sido modificados y GIT los está siguiendo.

- **git checkout -b NombreDeBranch**

Crea un nuevo branch y automáticamente GIT se cambia al branch creado, clonando el branch desde donde ejecutamos el comando.

- **git branch**

Nos muestra una lista de los branches que existen en nuestro repositorio.

- **git checkout NombreDeBranch**

Sirve para moverse entre branches, en este caso vamos al branch que indicamos en el comando.

- **git merge NombreDeBranch**

Hace un merge entre dos branches, en este caso la dirección del merge sería entre el branch que indiquemos en el comando, y el branch donde estemos ubicados.

- **git status**

Nos indica el estado del repositorio, por ejemplo cuales están modificados, cuales no están siendo seguidos por GIT, entre otras características.

- **git clone URL/name.git NombreProyecto**

Clona un proyecto de git en la carpeta NombreProyecto.

- **git push origin NombreDeBranch**

Este comando va a subir los archivos al repositorio remoto, específicamente al branch que indiquemos.

- **git pull origin NombreDeBranch**

Hace una actualización en nuestro branch local, desde un branch remoto que indicamos en el comando.

II. Objetivo

Organizar adecuadamente las clases según su funcionalidad o propósito bajo un namespace o paquete.

III. Introducción

Las clases de las bibliotecas estándar del lenguaje están organizadas en jerarquías de paquetes. Esta organización en jerarquías ayuda a que las personas encuentren clases particulares que requieren utilizar.

En Java, el código correspondiente a cualquier clase pública ha de estar

definida en un fichero independiente con extensión .java.

El nombre del fichero ha de coincidir con el nombre de la clase.

En ocasiones, en un fichero se pueden incluir varias clases si sólo una de ellas es pública (esto es, las demás son únicamente clases auxiliares que utilizamos para implementar la funcionalidad correspondiente a la clase pública).

Las clases que se utilizan para implementar manejadores de eventos en aplicaciones con interfaces gráficas de usuario.

Una vez compilada, una clase, sea pública o no, da lugar a un fichero con extensión .class en el que se almacenan los bytecodes correspondientes al código de la clase

IV. Desarrollo

Las actividades que se realizaron fueron:

- I. Conocer los tableros canvas
- II. Crear un proyecto en NetBeans y en la terminal
- III. Crear un jar en NetBeans y en la terminal

IV. Crear la documentación javadoc con NetBeans y con la terminal

V. Publicar esa documentación en github

V. Código fuente

❖ Act. 1

Se trata de un lienzo para diseñar modelos de un proyecto.

Este instrumento facilita poder comprender y trabajar desde un punto de vista integrado que entiende a lo que se va a querer realizar, ayudando a definir como un proyecto crea, genera y desarrolla sus objetivos a cumplir. Esta herramienta nos ayudará a definir nuestros objetivos a cumplir, cuales están completos y cuales necesitan una modificación hasta tener lo deseado



❖ Act. 2

➤ POOP6:

package poop6;

public class POOP6 {

```

public static void main(String[] args)
{
    System.out.println("Soy la clase
POOP6");
}
}

```

❖ Act. 3

```

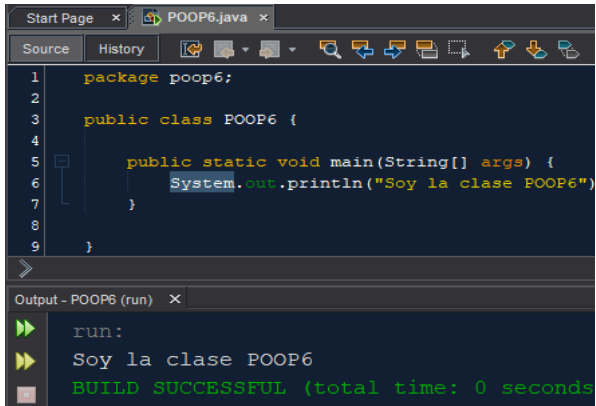
C:\Users\luisw\Desktop\POOP6V2>jar -cvfe POOP5.jar poop5.POOP5 poop5/*.class
added manifest
adding: poop5/Circulo.class(in = 1254) (out= 656)(deflated 47%)
adding: poop5/fecha.class(in = 1245) (out= 621)(deflated 50%)
adding: poop5/Persona.class(in = 1642) (out= 734)(deflated 55%)
adding: poop5/POOP5.class(in = 2894) (out= 1127)(deflated 46%)

```

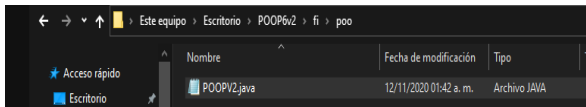
```

C:\Users\luisw\Desktop\POOP6V2>

```



➤ Terminal:



```

POOPV2.java: linea 4 de 4
Archivo Edición Formato Ver Ayuda
package fi.poo;

public class POOP6V2 {

    public static void main(String[] args) {
        System.out.println("Soy la clase POOP6V2")
    }

}

```

```

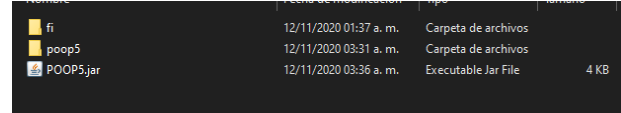
C:\Users\luisw\Desktop\POOP6V2>javac fi/poo/POOP6V2.j
C:\Users\luisw\Desktop\POOP6V2>java fi.poo.POOP6V2
Soy la clase POOP6V2
C:\Users\luisw\Desktop\POOP6V2>

```

```

C:\Users\luisw\Desktop\POOP6V2>javac poop5/POOP5.java
C:\Users\luisw\Desktop\POOP6V2>java poop5.POOP5
Circulo(radius=0.0, color=null)
Circulo(radius=6.0, color=null)
Perimetro c1 = 0.0
Area c2 = 113.09733552923255
1*****
Perimetro c1 = 56.548667764616276
Area c1 = 254.46900494077323
2*****
Info de mi Amigo Persona(Nombre=Edgar, Apellido=Romero Martinez, nacimiento=fecha(dia=3, mes=4, año= 2002))
Info de mi Amigo Persona(Nombre=Edgar, Apellido=Romero Martinez, nacimiento=fecha(dia=3, mes=4, año= 2002))
Info Amigo 2- Persona(Nombre=Raul, Apellido=Ayala Barbosa, nacimiento=fecha(dia=20, mes=4, año= 1993))
Info Amigo 3- Persona(Nombre=Diego, Apellido=Pedraza Peralta, nacimiento=fecha(dia=3, mes=5, año= 2000))
C:\Users\luisw\Desktop\POOP6V2>

```



```

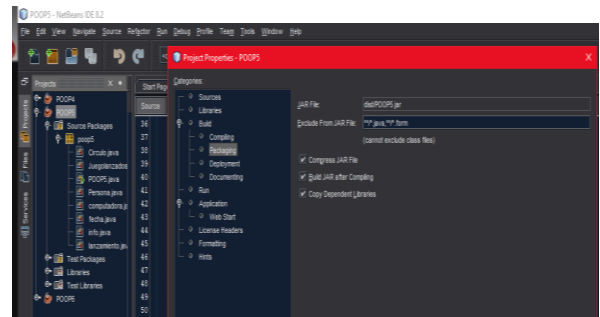
C:\Users\luisw\Desktop\POOP6V2>jar tf POOP5.jar
META-INF/
META-INF/MANIFEST.MF
poop5/Circulo.class
poop5/fecha.class
poop5/Persona.class
poop5/POOP5.class

```

```

C:\Users\luisw\Desktop\POOP6V2>java -jar POOP5.jar
Circulo(radius=0.0, color=null)
Circulo(radius=6.0, color=null)
Perimetro c1 = 0.0
Area c2 = 113.09733552923255
1*****
Perimetro c1 = 56.548667764616276
Area c1 = 254.46900494077323
2*****
Info de mi Amigo Persona(Nombre=Edgar, Apellido=Romero Martinez, nacimiento=fecha(dia=3, mes=4, año= 2002))
Info de mi Amigo Persona(Nombre=Edgar, Apellido=Romero Martinez, nacimiento=fecha(dia=3, mes=4, año= 2002))
Info Amigo 2- Persona(Nombre=Raul, Apellido=Ayala Barbosa, nacimiento=fecha(dia=20, mes=4, año= 1993))
Info Amigo 3- Persona(Nombre=Diego, Apellido=Pedraza Peralta, nacimiento=fecha(dia=3, mes=5, año= 2000))
C:\Users\luisw\Desktop\POOP6V2>

```



```

Output-POOP5 (clean.jar) X
ant -f "C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5" -Dch.internal.action.name=rebuild clean jar
Last:
log:clean:
Updating property file: C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\build\build-clean.properties
Deleting directory C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\build
Clean:
Last:
log:jar:
Created dir: C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\build
Updating property file: C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\build\build-jar.properties
Created dir: C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\build\classes
Created dir: C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\build\empty
Created dir: C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\build\generated-sources\ap-source-output
Compiling 8 source files to C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\build\classes
compile:
Created dir: C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\dist
Copying 1 file to C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\build
Working to copy.
Building jar: C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\dist\POOP5.jar
To run this application from the command line without Ant, try:
java -jar "C:\Users\luisw\Desktop\tareas UNAM\Tercer Semestre\POO\POOP5\dist\POOP5.jar"
JAR:
BUILD SUCCESSFUL (total time: 0 seconds)

```

Nombre	Fecha de modificación	Tipo	Tamaño
POOP5.jar	12/11/2020 03:52 a. m.	Executable Jar File	9 KB
README.TXT	12/11/2020 03:52 a. m.	Documento de te...	2 KB

Personalmente se me complico al realizarlo en la terminal pero dejando de lado ese problema siento que nos

podrá ayudar al realizar un gran proyecto y sin la necesidad de tener varios códigos.

- Francisco Moisés Barrera Guardia:

En esta práctica la verdad hubieron muchas complicaciones ya que hubo muchas cosas que no quedaron muy claras, aunque al final gracias a los ejercicios vistos, fueron un poco más claros, además, de que el uso de github es nuevo para mí, pude acoplarme un poco rápido a este nuevo modo de usar los repositorios y el guardar en una nube mis programas y así algún día trabajar más a fondo con ellos

VII. Referencias

- <https://victorrobesweb.es/2018/04/28/que-es-git-y-para-que-sirve/>
- <https://es.wikipedia.org/wiki/GitHub>
- <http://www.7sabores.com/blog/git-comandos-basicos>
- <https://community.canvaslms.com/t5/Deploys/Canvas-LMS-Github-Repository-Tutorial/ta-p/248021>
- <https://www.ticbeat.com/innovacion/modelo-canvas-aplicarlo-negocio/#:~:text=El%20modelo%20Canvas%20%2Dhabitualmente%20denominado,crea%2C%20genera%20y%20desarrolla%20valor.>

- <https://elvex.ugr.es/decsai/java/pdf/9C-packages.pdf>