



Bienvenidos!



00. Introducción a recursos de trabajo

Laboratorio de Redes de computadoras



Plan de Evaluación

Reglas del
juego

Laboratorio

- Participación en clase:
 - Una noticia sobre tecnología al inicio de la sesión.
 - Entrega a tiempo de reportes si son indicados
- Trabajo en equipo (4 personas).
- Respeto durante las actividades.
- Videos de las clases en el canal oficial de la materia.

Agenda del día

- Requisitos y Lenguajes de Programación
- Recursos de la Facultad de Ciencias
 - GitHub Student Developer Pack
 - AWS Educate para estudiantes
- Virtualización
- Introducción a GitLab
- Introducción a Sockets



Requisitos y Lenguajes de Programación



Conocimientos básicos de GNU/Linux



Conocimientos de Sistemas Operativos



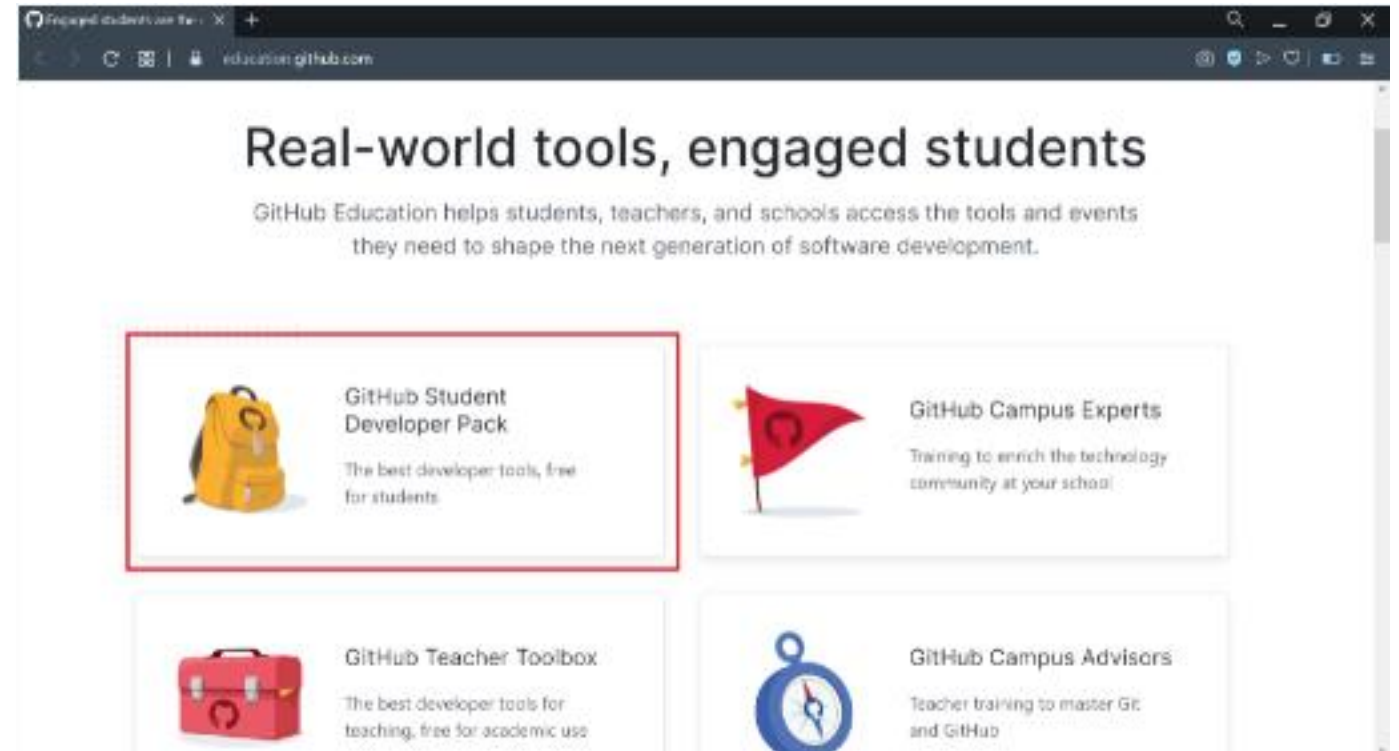
Conocimientos de programación en un lenguaje estructurado u orientado objetos



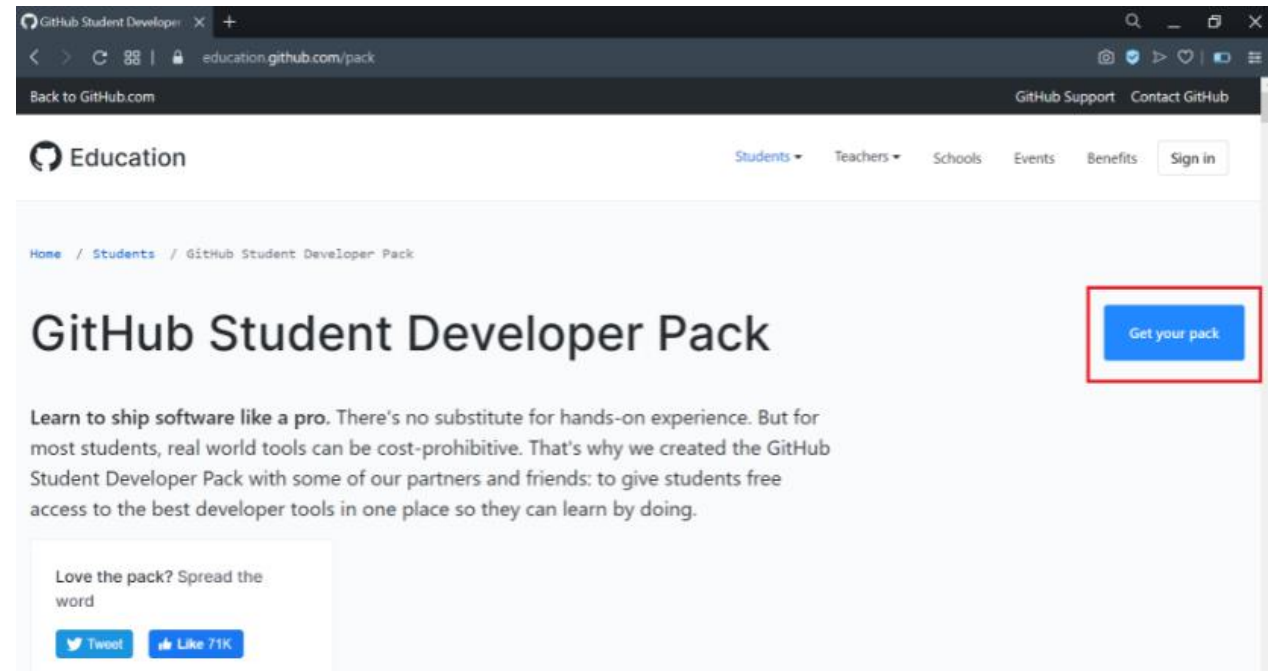
Se usara Python

GitHub Student Developer Pack

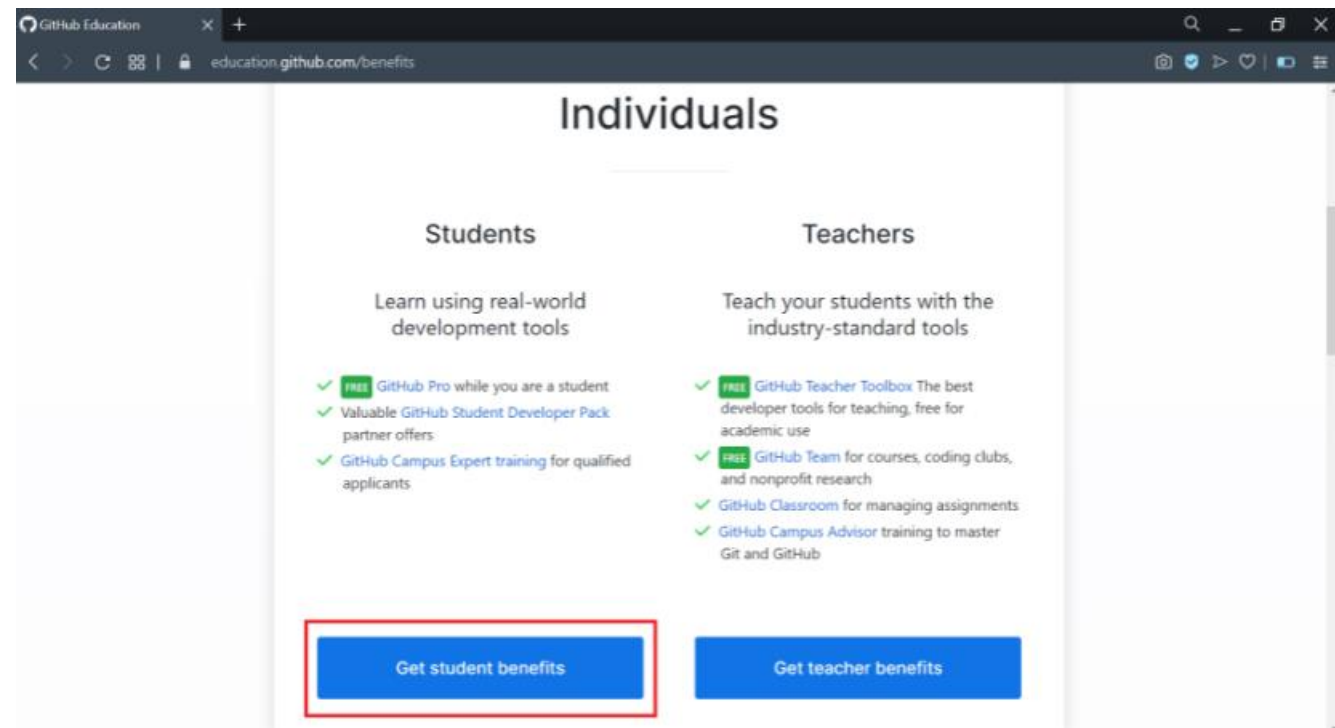
Acceder al sitio <https://education.github.com> y dar clic en el botón *GitHub Student Developer Pack*, para acceder a los beneficios del programa.



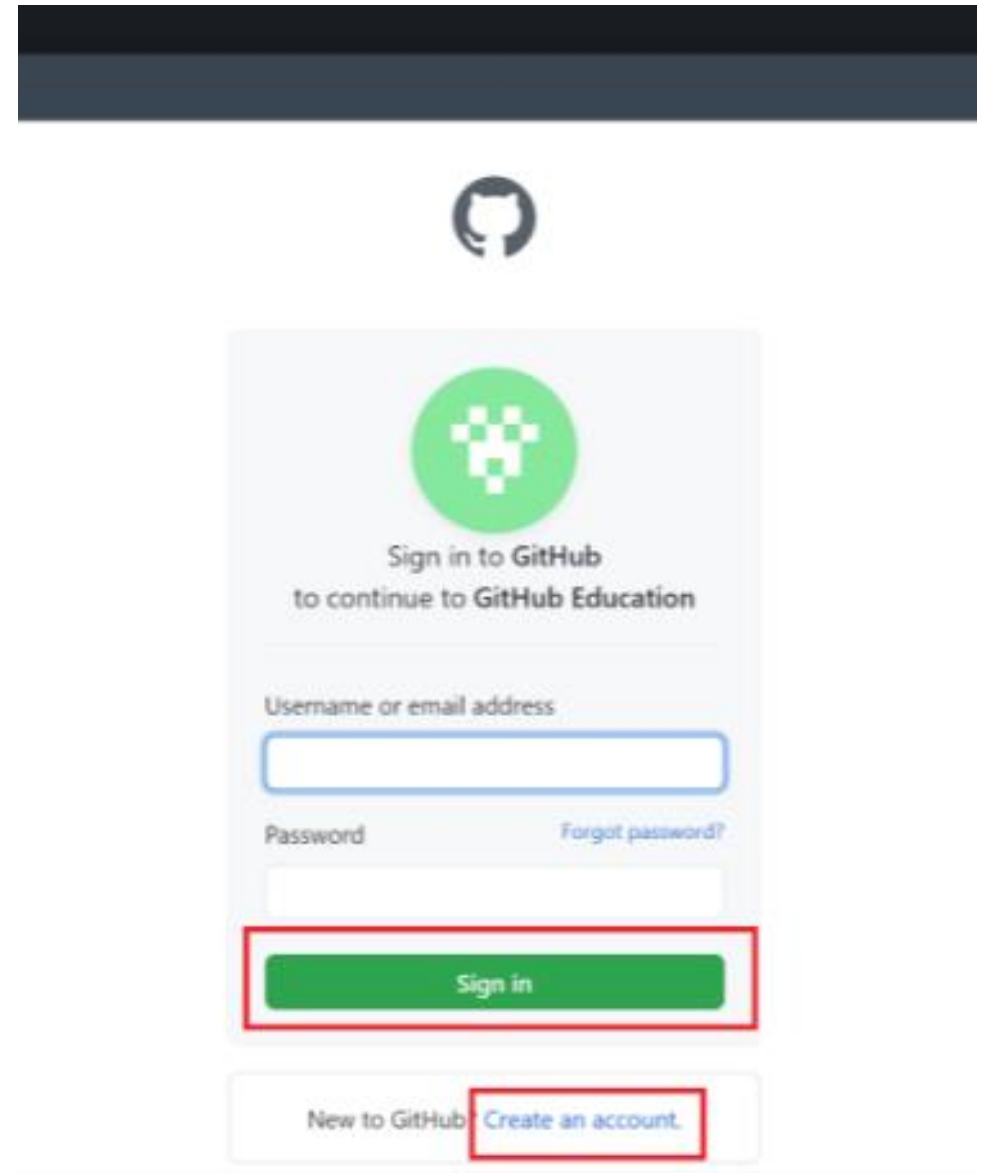
En la siguiente pantalla se pueden revisar todos los servicios y herramientas que ofrece el programa. Para continuar con el registro, dar clic en el botón *Get your pack*.



A continuación se muestran los beneficios del programa tanto para estudiantes como para profesores. Dar clic en el botón *Get student benefits*.



Para el siguiente paso es necesario ingresar usando una cuenta de GitHub, si ya se tiene una bastará con ingresar los datos de acceso que se solicitan; en caso de que no se tenga una cuenta, entonces dar clic en el enlace de *Create an account*, y complementar con los datos que se soliciten. Es necesario que el correo electrónico usado sea de un dominio @unam.mx, como por ejemplo, las cuentas de correo @ciencias.unam.mx, o @comunidad.unam.mx.



The image shows the GitHub login interface. At the top, there is a dark header bar. Below it, the GitHub logo (Octocat) is centered. The main content area has a light gray background. In the center, there is a green circular icon with a white pixelated pattern. Below this icon, the text "Sign in to GitHub" and "to continue to GitHub Education" is displayed. There are two input fields: "Username or email address" and "Password". To the right of the password field is a link "Forgot password?". Below the input fields is a green "Sign in" button, which is highlighted with a red rectangular border. At the bottom, there is a link "New to GitHub" followed by a "Create an account." link, which is also highlighted with a red rectangular border.

Una vez que se ha ingresado con una cuenta de GitHub, complementar el formulario de solicitud de ingreso al programa *Github Student Developer Pack*, se sugiere usar los datos que se muestran en la siguiente imagen. Si la cuenta de GitHub se creó con una cuenta que no es del dominio **@unam.mx**, entonces será necesario agregar una dirección extra de correo electrónico de dicho dominio, para tener mayor probabilidad de ser aceptados en el programa, ésto se hace a través del botón *Add an email address*. Una vez que se envían los datos, éstos no pueden ser modificados.

Ing.Dante Santiago Rodríguez P.

The screenshot shows the 'Request a discount' page on GitHub, specifically the 'education.github.com/discount_requests/student_application' form. The form is titled 'What e-mail address do you use for school?' and includes a note: 'Note: Selecting a school-issued email address gives you the best chance of a speedy review.' A red box highlights the email selection area, which shows a dropdown menu with 'National Autonomous University of Mexico - Universidad Nacional Autónoma de México (UNAM)' selected. Below this, another red box highlights the '+ Add an email address' button. The form then asks 'What is the name of your school?' with a note: 'Note: If your school is not listed, then enter the full school name and continue. You will be asked to provide further information about your school on the next page.' The school name 'National Autonomous University of Mexico - Unive' is entered in the text box. Below this, a third red box highlights the 'How do you plan to use GitHub?' section, which contains a text box with the answer: 'To develop a computer networking project. A code repository is required, and a domain for'. At the bottom, a green bar contains the text: 'Please note, your request cannot be edited once it has been submitted, so please verify your details for accuracy before sending them to us.'

- Una vez que se ha llenado y enviado la solicitud, verificar el correo electrónico que indica que ha sido aprobada la solicitud.

🌟 Welcome to the GitHub Student Developer Pack! 🌟



GitHub Education <education@github.com>

Lun 02/11/2020 09:23 PM

Para: PAULO [REDACTED]



GitHub Education

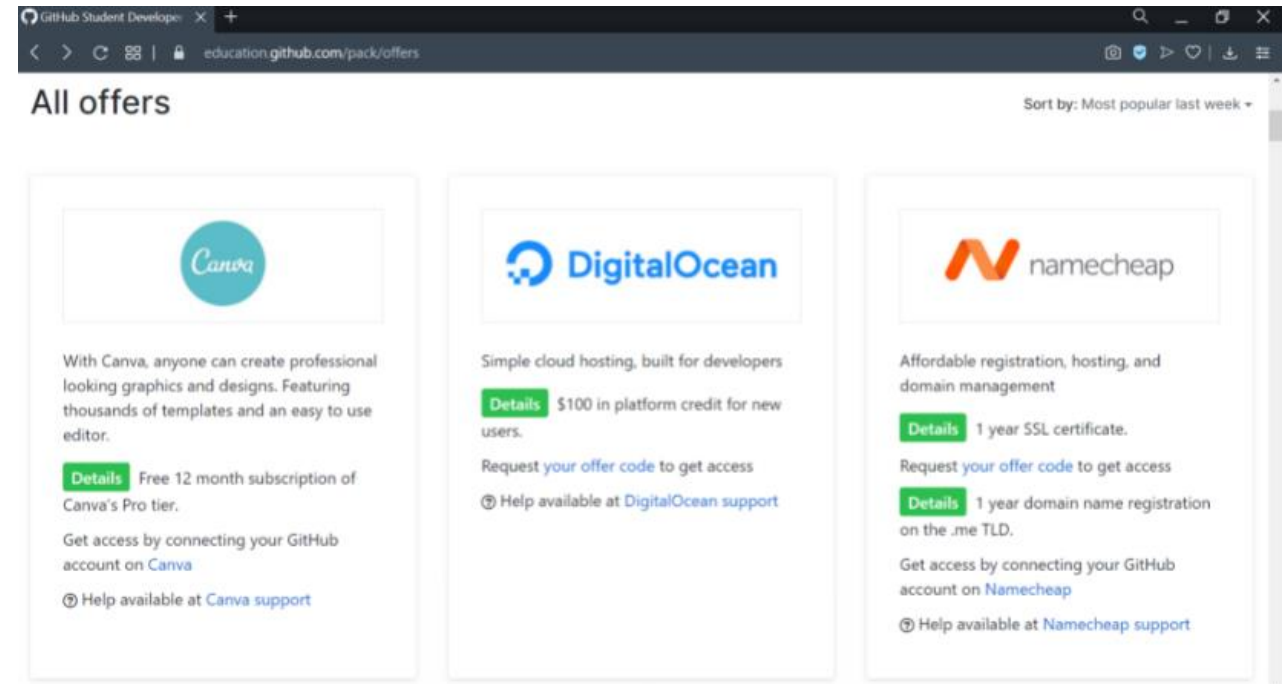
Hey @paulo[REDACTED]

Welcome! Thank you for joining [GitHub Education](#), we're so happy you're here! GitHub Education helps students, teachers, and schools access the tools and events they need to shape the next generation of software development.



GitHub
Student Developer Pack

- Para acceder a los recursos del programa, ingresar a <https://education.github.com/pack/offers> , se deberá de estar autenticado. Aquí se encontrarán los detalles sobre la forma de obtener los recursos de cada herramienta o servicio. Esta lista se actualiza constantemente.



AWS Educate para estudiantes

- La Facultad de Ciencias en conjunto con Amazon Web Services, pone a la disposición de sus estudiantes el programa *AWS Educate*, con el cual podrán obtener una cuenta educativa que les permitirá acceder a los recursos necesarios para su formación académica en tecnologías en la nube.



- Para consultar el manual para adherirse a este programa ingresar a https://computo.fciencias.unam.mx/guias_utilidad.html , y dar clic en Manual AWS Educate para estudiantes.
- Nota: Actualización de la liga de registro: https://www.awseducate.com/registration#APP_TYPE
- Se despliega en la parte inferior el manual en formato PDF o la versión en línea.





Virtualización

Virtualizar

- La virtualización es una tecnología probada de software que permite ejecutar múltiples sistemas operativos y aplicaciones simultáneamente en un mismo equipo.
- Es decir, convertir hardware en software, ocupando los recursos del equipo anfitrión de tal forma que un equipo físico parezca como varios lógicos.

Introducción a los sistemas de virtualización

- Un hipervisor o monitor de máquinas virtuales (VMM) es un programa que ejecuta y administra los recursos para las máquinas virtuales que corren dentro del servidor físico (host).
- Existen dos tipos de hipervisores que se diferencian de acuerdo al contexto donde se ejecutan:

Bare-Metal

- Este tipo de *hipervisores* se ejecuta directamente sobre el *hardware* del equipo físico.
- Como el *hipervisor* controla el *hardware* el rendimiento es mejor, pero el sistema únicamente se puede dedicar a correr máquinas virtuales.

Hosted

- Este tipo de *hipervisores* se ejecuta sobre un sistema operativo instalado en el equipo (GNU/Linux, MacOS X, FreeBSD, Solaris, Windows, etc.)
- Se ejecuta como una aplicación más en el sistema operativo
- El rendimiento es más lento, pero no es necesario dedicar el equipo únicamente a correr máquinas virtuales.

Tendencias en virtualización



Virtualización en la Nube

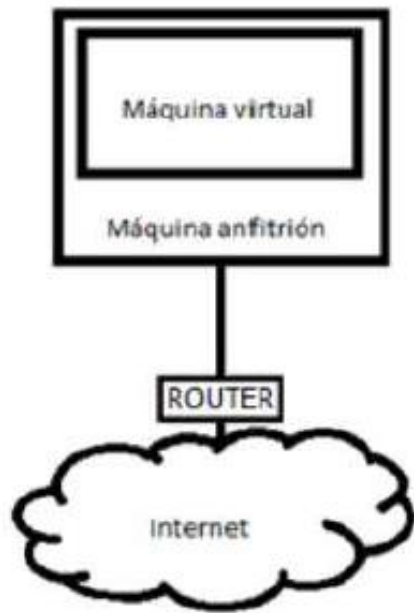
1. Amazon Web Services (AWS)
2. IBM PowerVM
3. Oracle VM
4. Windows Azure

Virtualización en Contenedores

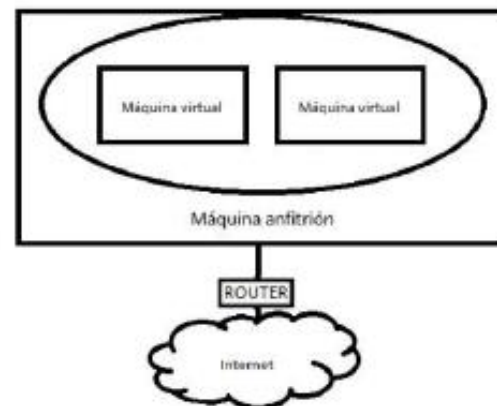
1. Docker
2. Kubernetes
3. LXC
4. Vagrant
5. Windows Container



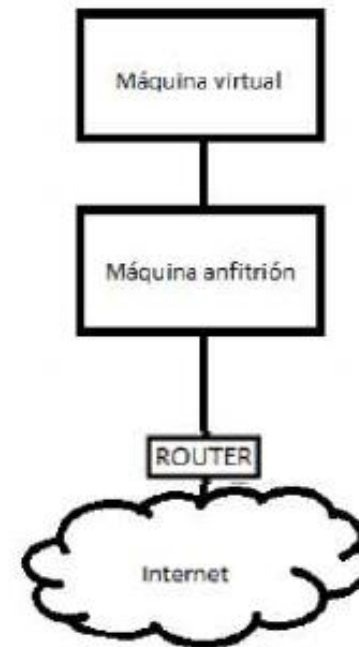
Tipos de conexión de red en VMware y VirtualBox



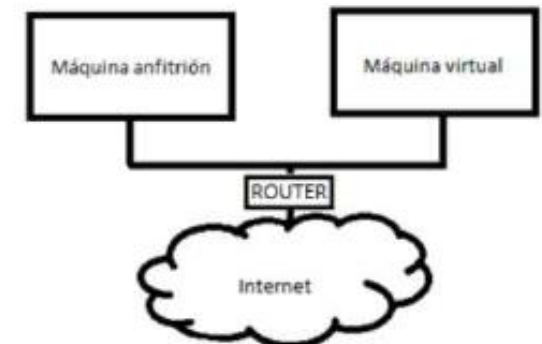
HOST-ONLY



RED INTERNA



NAT



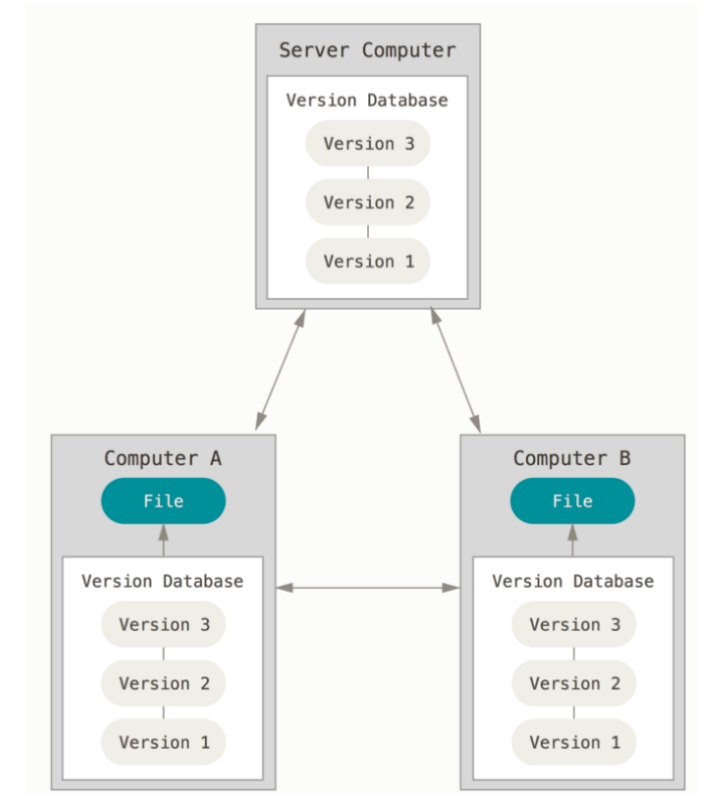
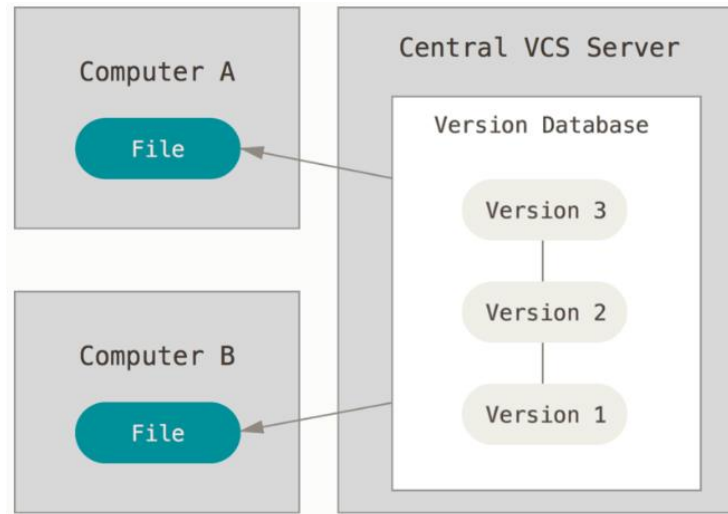
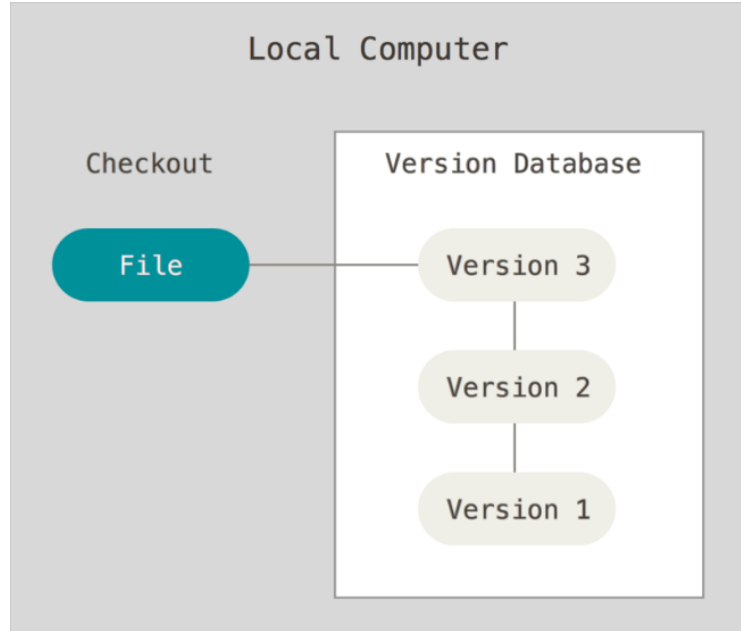
BRIDGE

Introducción a GitLab

- **Conocimientos previos**
 - Conocimientos básicos de GNU/Linux y Windows a nivel usuario
 - Uso de internet
 - Cuenta de correo electrónico
 - Conocimientos básicos de inglés a nivel técnico



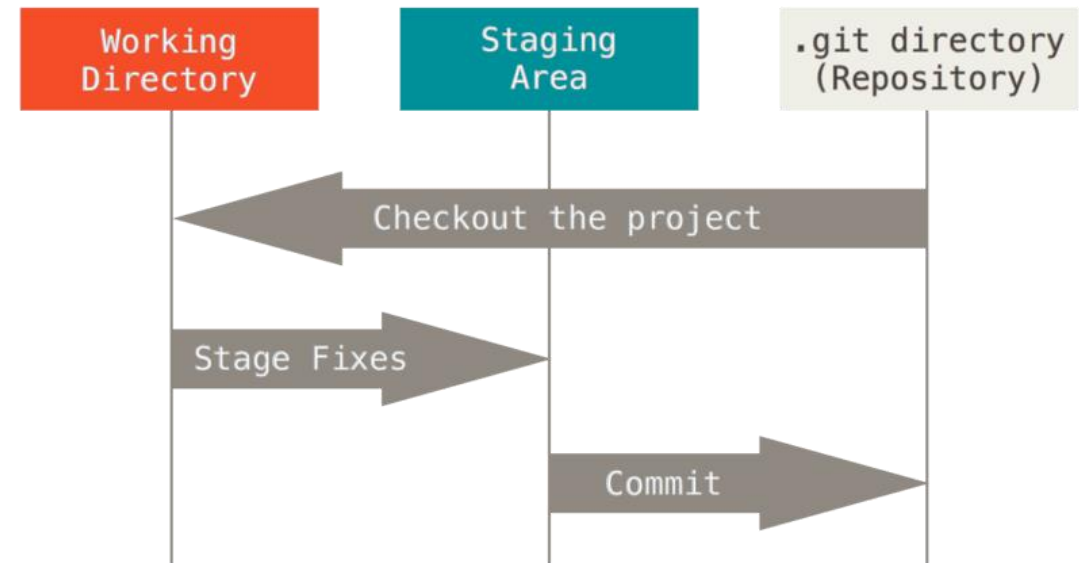
GitLab



Tipos de sistemas de control de versiones

Secciones principales

- **Working Directory** es el directorio de trabajo.
- En él se crean y borran los archivos del proyecto.
- **Staging Area** es el área donde se preparan los cambios que serán versionados.
- El directorio **.git** guarda todas las versiones de los archivos del proyecto.



Estado	Descripción	Comando
Modified	El archivo fue editado en el <i>directorio de trabajo</i>	<code>editor archivo</code>
Staged	El archivo (nuevo o modificado) fue agregado al <i>área de staging</i>	<code>git add archivo</code>
Staged	El archivo se movió o renombró utilizando <code>git</code>	<code>git mv archivo1 archivo2</code>
Staged	El archivo se borró del <i>área de staging</i> utilizando <code>git</code>	<code>git rm archivo</code>
Committed	Los cambios del archivo fueron guardados en el repositorio	<code>git commit archivo</code>

Estados

El flujo de trabajo básico en Git es:

1. Modificar archivos en tu directorio de loca.
2. Preparas los archivos, añadiéndolos a tu área de preparación.
3. Confirmas los cambios, lo que toma los archivos tal y como están en el área de preparación y almacena esa copia instantánea de manera permanente en tu directorio de Git.

Instalar git en GNU/Linux

```
root@KAME-HOUSE:~# git --version
git version 2.30.2
root@KAME-HOUSE:~#
```

```
root@KAME-HOUSE:~# aptitude install git
The following NEW packages will be installed:
  ca-certificates{a} git git-man{a} libbrotli1{a} libcurl3-gnutls{a} liberror-perl{a} libexpat1{a} libldap-2.4-2{a}
  libldap-common{a} libnghttp2-14{a} libpsl5{a} librtmp1{a} libsasl2-2{a} libsasl2-modules{a}
  libsasl2-modules-db{a} libssh2-1{a} openssl{a} patch{a} publicsuffix{a}
0 packages upgraded, 19 newly installed, 0 to remove and 0 not upgraded.
Need to get 10.3 MB of archives. After unpacking 44.4 MB will be used.
Do you want to continue? [Y/n/?]
Get: 1 http://security.debian.org/debian-security bullseye-security/main amd64 openssl amd64 1.1.1k-1+deb11u1 [851 kB]
```

- En Debian GNU/Linux utilizando apt-get o aptitude
root@KAME-HOUSE:~# aptitude install git
- En Red Hat, CentOS y Oracle Linux se utiliza yum
[root@Kame-House~]# yum install git
- Verificar versión instalada
root@KAME-HOUSE:~# git --version
- También existen ambientes para Windows y MAC con interfaz grafica como *gitKraken*.

Terminal



Comando	Descripción
git init	Inicializar un nuevo repositorio
git status	Checa el estatus del repositorio actual
git add nombre_archivo git add.	Con el nombre del archivo se registra dicho cambio en el caso de alguna creación, con el punto (.) se preparan todos los cambios realizados en el proyecto actual. Lista de cambios preparados para subir en el siguiente commit.
git commit git commit -m 'initial commit of my project'	Genera un archivo de los cambios realizados en el proyecto, en el archivo que se mostrará se agrega una descripción del cambio realizado.
git push --set-upstream NAME_REPO master git push NAME_REPO BRANCH_NAME git push REMOTE --all git push REMOTE --tags	Subir cambios de un Proyecto a gitLab
git clone URL	Se utiliza para guardar una copia de un repositorio existente
git branch	Revisa que de manera inicial te encuentres en la rama
git branch -a	Muestra la lista de las ramas creadas
git checkout -b xxxxxxxx	Creación de una nueva rama y ubicación dentro de esa rama

GitLab

- Crear cuenta en gitlab
- Se generará un repositorio de entrega de prácticas y tareas. Posteriormente les daremos dicho enlace.

GitLab.com

GitLab.com offers free unlimited (private) repositories and unlimited collaborators.

- [Explore projects on GitLab.com](#) (no login needed)
- [More information about GitLab.com](#)
- [GitLab Community Forum](#)
- [GitLab Homepage](#)

By signing up for and by signing in to this service you accept our:

- [Privacy policy](#)
- [GitLab.com Terms](#)

Username or email

Password

☐ Remember me [Forgot your password?](#)

Sign in

Don't have an account yet? [Register now](#)

Sign in with

Google


GitHub

GitLab

Menu

Search GitLab

Sign in / Register



Dante Erik Santiago Rodríguez Pérez

@SantiagoDantes · Member since September 22, 2021

9:36 PM

0 followers · 1 following

Overview

Activity

Groups

Contributed projects

Personal projects

Starred projects

Snippets

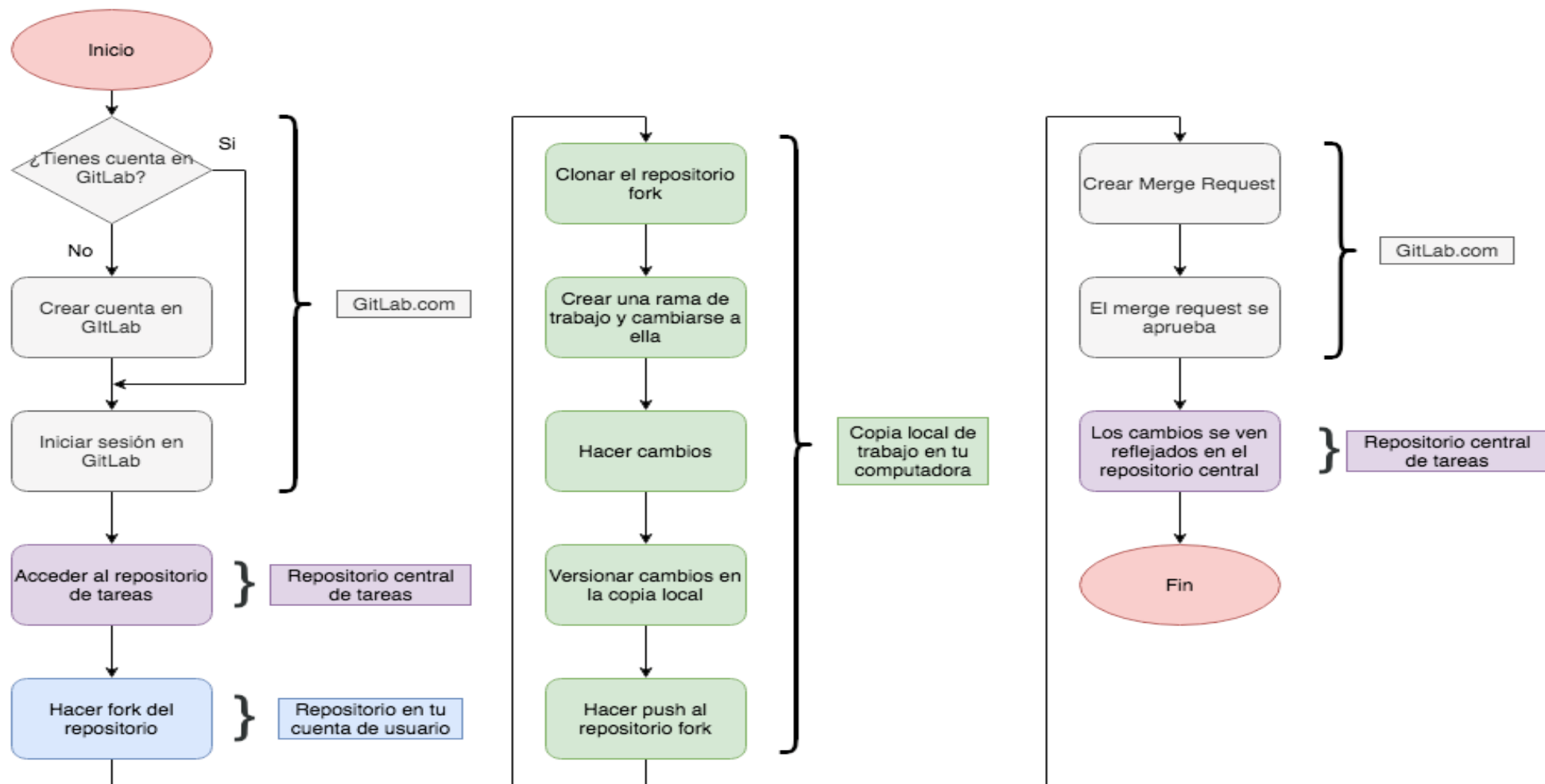
Followers

Following

	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
M													
W													
F													

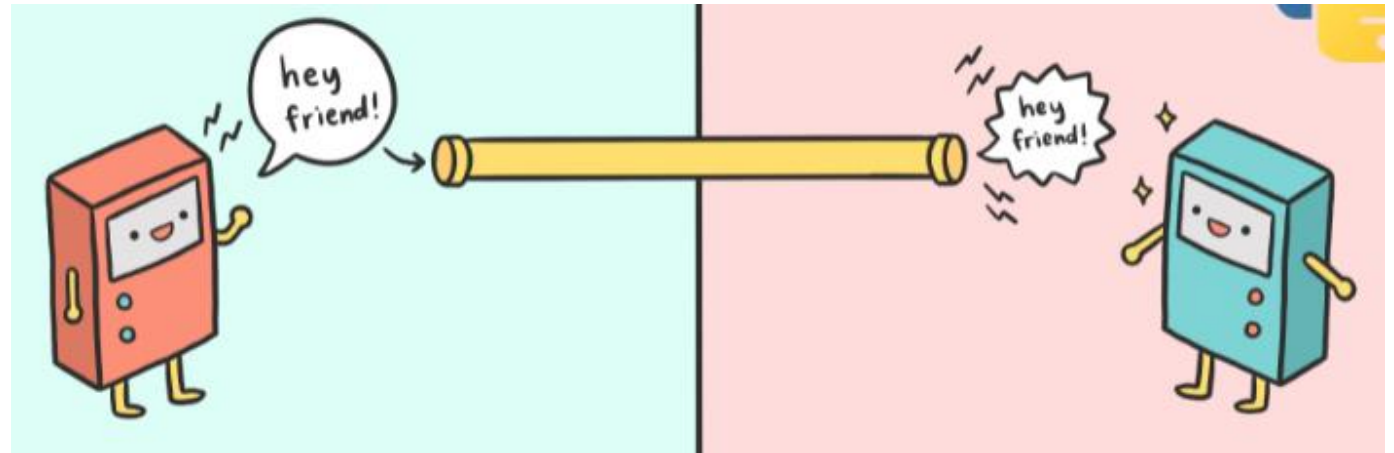
Issues, merge requests, pushes, and comments.

Flujo de trabajo para la entrega de tareas y prácticas



Comunicación (SOCKETS)

- Es una abstracción de una comunicación bidireccional (en ocasiones llamada asociación) entre dos procesos.
- Son una “Application Programming Interface (API)” para comunicación interprocesos (IPC)
- A pesar de no ser un estándar, se ha convertido en un mecanismo de facto en la industria.



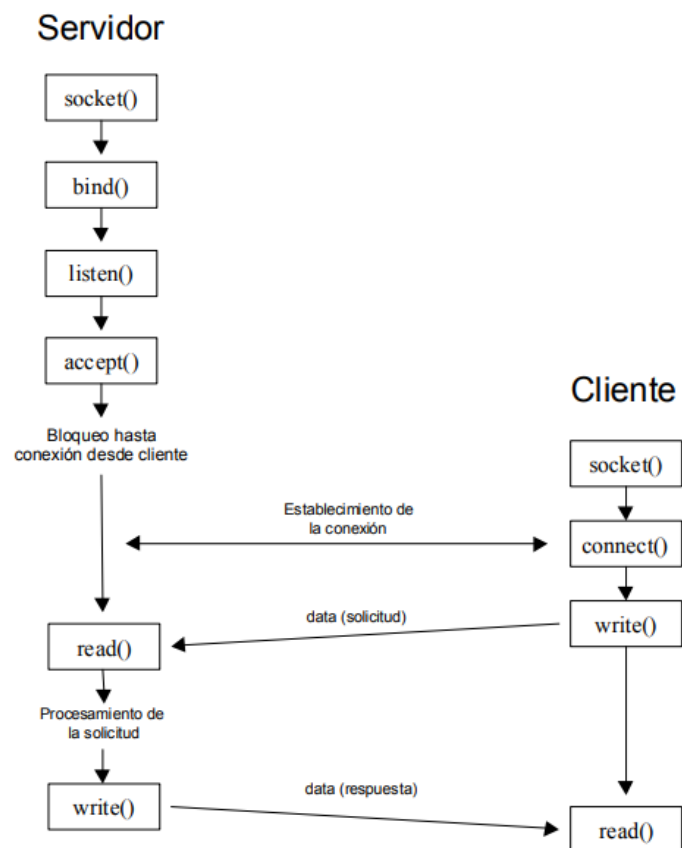
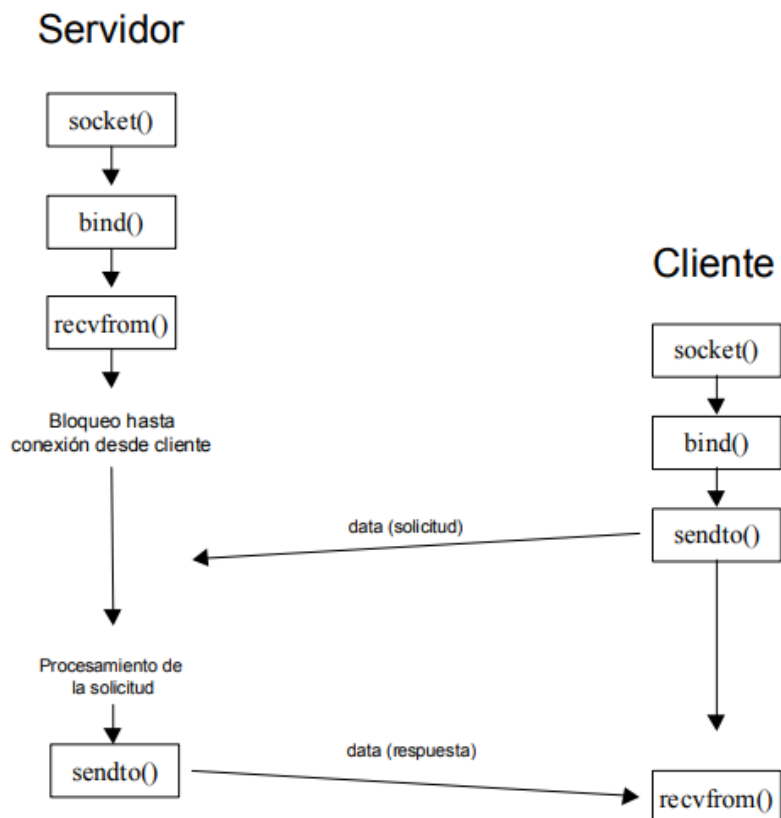
Familias de protocolos

- PF_INET:IPv4 (direcciones de 32 bits)
- PF_INET6: IPv6 (direcciones de 64 bits)
- PF_UNIX: Mecanismos IPC en un mismo equipo
- PF_APPLETALK: Redes Applestalk
- PF_IPX: Redes Novel Netware
- PF_ALG:API criptográfica del Kernel Linux
- Etc..


- SOCK_STREAM: Orientado a conexión (TCP)-Flujo
- SOCK_DGRAM: No orientado a conexión (UDP)-Datagramas
- SOCK_RAW: Acceso directo a la capa de red (ICMP)
- Socket_PACKET: Acceso directo a la capa de enlace
- Etc

Tipos de Sockets

Tipos de socket



API: Sockets de Berkeley/POSIX

- Permite
 - Configuración de direcciones
 - Creación de sockets “Bindig”, escucha
 - Inicialización y aceptación de una conexión
 - Envío y recepción de información
 - Destrucción de sockets
 - Técnicas de programación (monitoreo y tratamiento de errores)
- 
- A large yellow right-angled triangle is positioned in the bottom right corner of the slide, pointing towards the top right.

Empleando la API (En C)

- `int socket(int dominio, int tipo, int protocolo)`
 - dominio: Especifica la familia de protocolos(PF_INET)
 - tipo: Especifica el tipo socket (DGRAM-STREAM)
 - protocolo: Especifica el protocolo a ser usado por el socket (0 para default)

```

(root@kali)-[/home/kali/FCiencias]
# netstat -natup
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1095/sshd: /usr/sbi
tcp        0      0 127.0.0.1:6010          0.0.0.0:*               LISTEN      134329/sshd: kali@p
tcp        0      0 127.0.0.1:9050          0.0.0.0:*               LISTEN      1187/tor
tcp        0      0 127.0.0.1:6011          0.0.0.0:*               LISTEN      134391/sshd: kali@p
tcp        0      0 127.0.0.1:6012          0.0.0.0:*               LISTEN      134508/sshd: kali@p
tcp        0      0 127.0.0.1:6013          0.0.0.0:*               LISTEN      135193/sshd: kali@p
tcp        0      0 132.248.58.77:41360     52.38.198.132:443       ESTABLISHED 12639/x-www-browser
tcp        0      0 132.248.58.77:22        132.248.58.199:60765    ESTABLISHED 134501/sshd: kali [
tcp        0      0 132.248.58.77:22        132.248.58.199:36663    ESTABLISHED 135194/sshd: kali [
tcp        0      0 132.248.58.77:22        132.248.58.199:21908    ESTABLISHED 134386/sshd: kali [
tcp        0      0 132.248.58.77:22        132.248.58.199:61116    ESTABLISHED 134384/sshd: kali [
tcp        0      0 132.248.58.77:22        132.248.58.199:13069    ESTABLISHED 134503/sshd: kali [
tcp        0      0 132.248.58.77:37762     52.42.128.29:443       ESTABLISHED 12639/x-www-browser
tcp        0      0 132.248.58.77:22        132.248.58.199:24489    ESTABLISHED 134327/sshd: kali [
tcp        0      0 132.248.58.77:22        132.248.58.199:21801    ESTABLISHED 134322/sshd: kali [
tcp        0    256 132.248.58.77:22        132.248.58.199:6071    ESTABLISHED 135187/sshd: kali [
tcp6       0      0 :::22                  :::*                    LISTEN      1095/sshd: /usr/sbi
tcp6       0      0 :::1:6010              :::*                    LISTEN      134329/sshd: kali@p
tcp6       0      0 :::1:6011              :::*                    LISTEN      134391/sshd: kali@p
tcp6       0      0 :::1:6012              :::*                    LISTEN      134508/sshd: kali@p
tcp6       0      0 :::1:6013              :::*                    LISTEN      135193/sshd: kali@p

```

Visualizar Socket activos en el sistema

- A continuación se listan los comandos que permiten visualizar los sockets y sus estados:
 - #netstat (también esta Windows)
 - #ss

Ejemplo en python

Server

```
import socket
mi_socket = socket.socket()
mi_socket.bind ( 'localhost', 10000 )
mi_socket.listen(5)
while True:
    conexion, addr = mi_socket.accept()
    print "Nueva conexion exitosa"
    print addr
    peticion = conexion.recv(1024)
    print peticion
    conexion.send("Bienviedo al servidor de Redes")
    conexion.close()
```

```
(kali@kali)-[~/FCiencias]
$ python server.py
nueva conexion exitosa
('127.0.0.1', 52550)
nueva conexion exitosa
('127.0.0.1', 52552)
nueva conexion exitosa
('127.0.0.1', 52554)
```

Cliente

```
import socket

mi_socket = socket.socket()
mi_socket.connect(('localhost',10000))

mi_socket.send("Gracias por la bienvenida")
respuesta = mi_socket.recv(1024)

print respuesta
mi_socket.close()
```

```
(kali@kali)-[~/FCiencias]
$ python cliente.py
Bienviedo al servidor de Redes
```

Captura de tráfico

```
(root@kali)-[~]
# tcpdump -nnvXi lo
tcpdump: listening on lo, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:13:37.061072 IP (tos 0x0, ttl 64, id 6165, offset 0, flags [DF], proto TCP (6), length 60)
    127.0.0.1.52554 > 127.0.0.1.10000: Flags [S], cksum 0xfe30 (incorrect -> 0x3a74),
    seq 1861619038, win 65495, options [mss 65495,sackOK,TS val 860329736 ecr 0,nop,wscale
    7], length 0
    0x0000: 4500 003c 1815 4000 4006 24a5 7f00 0001 E..<..@.@.$.....
    0x0010: 7f00 0001 cd4a 2710 6eec e7f6 0000 0000 .....J'.n.....
    0x0020: a002 ffd7 fe30 0000 0204 ffd7 0402 080a .....0.....
    0x0030: 3347 9708 0000 0000 0103 0307 3G.....
21:13:37.061082 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    127.0.0.1.10000 > 127.0.0.1.52554: Flags [S.], cksum 0xfe30 (incorrect -> 0x740a),
    seq 1861619038, ack 1861619038, win 65483, options [mss 65495,sackOK,TS val 860329736
    ecr 860329736,nop,wscale 7], length 0
    0x0000: 4500 003c 0000 4000 4006 3cba 7f00 0001 E..<..@.@.<.....
    0x0010: 7f00 0001 2710 cd4a 5366 a8ae 6eec e7f7 ....'.JSf..n...
    0x0020: a012 ffc3 fe30 0000 0204 ffd7 0402 080a .....0.....
    0x0030: 3347 9708 3347 9708 0103 0307 3G..3G.....
21:13:37.061091 IP (tos 0x0, ttl 64, id 6166, offset 0, flags [DF], proto TCP (6), length 52)
    127.0.0.1.52554 > 127.0.0.1.10000: Flags [.], cksum 0xfe28 (incorrect -> 0x9ac6),
    ack 1, win 512, options [nop,nop,TS val 860329736 ecr 860329736], length 0
    0x0000: 4500 0034 1816 4000 4006 24ac 7f00 0001 E..4..@.@.$.....
    0x0010: 7f00 0001 cd4a 2710 6eec e7f7 5366 a8af ....J'.n...Sf..
    0x0020: 8010 0200 fe28 0000 0101 080a 3347 9708 .....(.....3G..
    0x0030: 3347 9708 3G..
21:13:37.061195 IP (tos 0x0, ttl 64, id 36275, offset 0, flags [DF], proto TCP (6), length 82)
    127.0.0.1.10000 > 127.0.0.1.52554: Flags [P.], cksum 0xfe46 (incorrect -> 0x2400),
    seq 1:31, ack 1, win 512, options [nop,nop,TS val 860329736 ecr 860329736], length 30
    0x0000: 4500 0052 8db3 4000 4006 aef0 7f00 0001 E..R..@.@.....
    0x0010: 7f00 0001 2710 cd4a 5366 a8af 6eec e7f7 ....'.JSf..n...
    0x0020: 8018 0200 fe46 0000 0101 080a 3347 9708 .....F.....3G..
    0x0030: 3347 9708 4269 656e 7669 6564 6f20 616c 3G..Bienviedo.al
    0x0040: 2073 6572 7669 646f 7220 6465 2052 6564 .servidor.de.Red
    0x0050: 6573 es
21:13:37.061209 IP (tos 0x0, ttl 64, id 36276, offset 0, flags [DF], proto TCP (6), length 52)
    127.0.0.1.10000 > 127.0.0.1.52554: Flags [F.], cksum 0xfe28 (incorrect -> 0x9aa7),
    seq 31, ack 1, win 512, options [nop,nop,TS val 860329736 ecr 860329736], length 0
    0x0000: 4500 0034 8db4 4000 4006 af0d 7f00 0001 E..4..@.@.....
    0x0010: 7f00 0001 2710 cd4a 5366 a8cd 6eec e7f7 ....'.JSf..n...
    0x0020: 8011 0200 fe28 0000 0101 080a 3347 9708 .....(.....3G..
    0x0030: 3347 9708 3G..
21:13:37.061246 IP (tos 0x0, ttl 64, id 6167, offset 0, flags [DF], proto TCP (6), length 52)
    127.0.0.1.52554 > 127.0.0.1.10000: Flags [.], cksum 0xfe28 (incorrect -> 0x9aa8),
    ack 31, win 512, options [nop,nop,TS val 860329736 ecr 860329736], length 0
    0x0000: 4500 0034 1817 4000 4006 24ab 7f00 0001 E..4..@.@.$.....
    0x0010: 7f00 0001 cd4a 2710 6eec e7f7 5366 a8cd ....J'.n...Sf..
    0x0020: 8010 0200 fe28 0000 0101 080a 3347 9708 .....(.....3G..
    0x0030: 3347 9708 3G..
21:13:37.061270 IP (tos 0x0, ttl 64, id 6168, offset 0, flags [DF], proto TCP (6), length 77)
    127.0.0.1.52554 > 127.0.0.1.10000: Flags [S.], cksum 0xfe30 (incorrect -> 0x740a),
    seq 1861619038, ack 1861619038, win 65483, options [mss 65495,sackOK,TS val 860329736
    ecr 860329736,nop,wscale 7], length 0
    0x0000: 4500 003c 0000 4000 4006 3cba 7f00 0001 E..<..@.@.<.....
    0x0010: 7f00 0001 2710 cd4a 5366 a8ae 6eec e7f7 ....'.JSf..n...
    0x0020: a012 ffc3 fe30 0000 0204 ffd7 0402 080a .....0.....
    0x0030: 3347 9708 3347 9708 0103 0307 3G..3G.....
```

Para entregar

1. Si no se cuenta con cuenta de gitlab, generarla.
2. Se deberá de crear un programa *cliente.c* y un programa *servidor.c* . Ambos programas deben de comunicarse a través de sockets, wel *servidor.c* debe recibir un argumento que indicará el puerto por el que se pondrá en escucha, *cliente.c* recibirá dos argumentos, la dirección IPv4 del servidor al que se quiere conectar, y el puerto. El cliente al conectarse al servidor, deberá de poder enviar comandos que serán ejecutados por el servidor, la salida de estos comandos será entonces enviada al cliente. También se debe de enviar la salida de error estándar.

Python

- Crear un programa que lea un archivo que tenga el formato de `/etc/passwd` y muestre el significado de los campos de cada usuario. Debe de leer el nombre del archivo desde la línea de comandos.

Ejemplo

Si `/etc/passwd` tiene el siguiente contenido:

```
root:x:0:0:root:/root:/bin/bash
ricardo::1000:1000:Usuario ricardo:/home/ricardo:/bin/sh
ramon:$1$kmIB05Wr$s2Ty91SbBvFJQLaGqc/LG/:1001:1001:Usuario ramon:/home/ramon:/bin/sh
ruben:$5$qbgHgwa6q7yGUHF$9G6iaR/qGUoENb1AczFoFvzovEYA6rMIFxn8ljtjia5:1002:1002:/bin/sh
```

Entonces el programa debe de tener la siguiente salida:

```
[usr@srv]$ ./lee_passwd /etc/passwd
Nombre de usuario: root
Contraseña: Contraseña en archivo /etc/shadow
UID: 0
GID: 0
GECOS: root
Home: /root
Shell: /bin/bash

Nombre de usuario: ricardo
Contraseña: Autenticación sin contraseña
UID: 1000
GID: 1000
GECOS: Usuario ricardo
Home: /home/ricardo
Shell: /bin/sh
```

Adicional

- Usar el virtualizador de su preferencia para crear un laboratorio con dos maquinas virtuales para realizar algunas prácticas.
- Se ocuparan de 2 a 3 maquinas virtuales con los sistemas Linux (Debian y Kali) y Windows.



Referencia

Redes de Computadoras - Facultad de Ciencias - UNAM

- <http://redes-ciencias-unam.gitlab.io/laboratorio/practica0/>

Git --everything-is-local

- <https://git-scm.com/book/es/v2>

Control de versiones con git

- <https://tonejito.github.io/curso-git/book.html>