

# Sistemas Operativos 1

## Introducción

Edwin Salvador

15 de abril de 2015

Sesión 2

# Contenido I

## 1 Control de versiones

- Github
- Git
- Repositorios
- add, commit, push y otros comandos

## 2 Virtualización

- Virtualbox

## 3 Ubuntu

- Requerimientos
- Instalación

## 4 Introducción

- El computador

# Control de versiones con Git

- Git es un sistema de control de versiones (VCS) de código abierto y distribuido capaz de trabajar con proyectos grandes y pequeños de manera eficiente.
- Git funciona estrictamente mediante la línea de comandos.
- Git facilita la interacción con *GitHub* (un repositorio web con una interfaz amigable).
- Brevemente veamos como instalar y configurar Git en nuestro ambiente de trabajo.

# Contenido I

## 1 Control de versiones

- Github
- Git
- Repositorios
- add, commit, push y otros comandos

## 2 Virtualización

- Virtualbox

## 3 Ubuntu

- Requerimientos
- Instalación

## 4 Introducción

- El computador

# Creando un repositorio remoto en GitHub

- Clic en el (+) en la esquina superior derecha (Nuevo repositorio).
- Ingresar nombre del repositorio (esfot\_ambientesnopropietarios)
- Ingresar descripción del repositorio
- Seleccionar Público, no inicializar con README.md, .gitignore = none, license = none.
- Clic en crear

# Contenido I

## 1 Control de versiones

- Github
- **Git**
- Repositorios
- add, commit, push y otros comandos

## 2 Virtualización

- Virtualbox

## 3 Ubuntu

- Requerimientos
- Instalación

## 4 Introducción

- El computador

# Instalando Git

- Crear una cuenta gratuita en <http://github.com>
- Descargar Git bash desde:
  - <http://git-scm.com> (**Trabajaremos con esta**)
- GitHub ofrece una interfaz gráfica (no necesario):
  - <https://windows.github.com> (Windows)
  - <https://mac.github.com> (Mac)

# Configurando Git

Empezamos con la configuración del usuario para todos los repositorios locales.

- El nombre que queremos que vaya con nuestras transacciones commit:  
`$ git config --global user.name "[nombre]"`
- El email que queremos que vaya con nuestras transacciones commit:  
`$ git config --global user.email "[email address]"`
- Habilitar colores en la línea de comandos:  
`$ git config --global color.ui auto`



# Contenido I

- 1 Control de versiones
  - Github
  - Git
  - Repositorios
  - add, commit, push y otros comandos
- 2 Virtualización
  - Virtualbox
- 3 Ubuntu
  - Requerimientos
  - Instalación
- 4 Introducción
  - El computador

# Creando repositorios

Para crear un nuevo repositorio o obtener uno desde una URL ya existente.

- Crear un nuevo repositorio local con un nombre específico:

```
$ git init [project-name]
```

- También podemos inicializar en un directorio ya existente, en este caso ejecutaremos solamente `git init`.
- Descargar un proyecto existente:

```
$ git clone [url]
```

# Configurando el repositorio local

- Crear archivo README para el repositorio:

```
touch README.md
```

- Modificar el archivo README creado

```
echo texto descriptivo del repositorio » README.md
```

- Crear .gitignore

```
touch .gitignore
```

- Modificar y añadir los archivos que se desean excluir del versionamiento. (\*.pdf, \*.tex, etc)

# Contenido I

- 1 Control de versiones
  - Github
  - Git
  - Repositorios
  - add, commit, push y otros comandos
- 2 Virtualización
  - Virtualbox
- 3 Ubuntu
  - Requerimientos
  - Instalación
- 4 Introducción
  - El computador

# Realizar cambios

Revisar las ediciones y empezar una transacción `commit`.

- Listar todos los archivos nuevos o modificados que necesitan ser subidos:

```
$ git status
```

- Mostrar las modificaciones en los archivos que no han sido subidas:

```
$ git diff
```

- Añadir un archivo al control de versiones:

```
$ git add [archivo]
```

- Mostrar las diferencias entre el archivo a subir y la última versión subida:

```
$ git diff --staged
```

- Quitar el archivo del control de versiones pero conservar su contenido:

```
$ git reset [archivo]
```

- Subir el archivo y los cambios al historial de versiones permanentemente:

```
$ git commit -m "[mensaje descriptivo]"
```

# Subir los cambios

- Vincular con el repositorio remoto (en GitHub)

```
git remote add origin https://github.com/NOMBRE_DE_USUARIO/NOMBRE_DEL_REPOSITORIO.git
```

- Subir los cambios al repositorio remoto

```
git push -u origin master
```

- INGRESAR USUARIO
- INGRESAR CONTRASEÑA
- Para evitar que nos pida usuario y contraseña cada vez, seguimos estos pasos:

<https://help.github.com/articles/generating-ssh-keys/>

# Modificando nombres de archivos

Mover o eliminar archivos con control de versiones.

- Eliminar un archivo del directorio actual y registrar la eliminación.

```
$ git rm [archivo]
```

- Eliminar el archivo del control de versiones pero mantener la copia local.:

```
$ git rm --cached [archivo]
```

- Cambiar el nombre del archivo o moverlo a otro directorio:

```
$ git mv [archivo-original] [archivo-renombrado]
```

# Contenido I

## 1 Control de versiones

- Github
- Git
- Repositorios
- add, commit, push y otros comandos

## 2 Virtualización

- Virtualbox

## 3 Ubuntu

- Requerimientos
- Instalación

## 4 Introducción

- El computador



# ¿Qué es la virtualización?

- Actualmente el desafío más importante de TI es el crecimiento de la infraestructura.
- Esto obliga a destinar el 70 % del presupuesto de Ti al mantenimiento dejando pocos recursos para la innovación.
- Los servidores de hoy están diseñados para ejecutar un solo SO.
- Debido a esto incluso pequeñas empresas se ven obligadas a obtener más de un servidor y cada uno será utilizado entre el 5 % y 15 % de su capacidad. Altamente ineficiente!
- La **virtualización** permite tener varios SO y aplicaciones en un solo servidor físico o anfitrión (*host*).
- Las **máquinas virtuales** (*VM, Virtual Machine*) están separadas de las otros y utiliza recursos hardware del host.

# Requerimientos mínimos para virtualización

- **CPU**

- Mínimo: Dual core (single socket)
- Ideal: 4 o más cores por CPU (Dual socket)

- **Memoria**

- Mín: 2GB
- Ideal: 8+GB

- **Red**

- Mín: 1 NIC
- Ideal: 1 para cada VM

- **Almacenamiento:**

- Local (SATA/SAS)
  - Mín: uno de 8GB
  - Ideal: 4 RAID5 por VM

- Aumento de hasta el 80 % en utilización de cada servidor.
- Menos requisitos de hardware en una proporción de 10:1 o superior.
- Reducción de los gastos operacionales y de capital a la mitad (ahorro de aprox \$1500 por cada servidor virtualizado).
- Alta disponibilidad sólida y rentable.
- Ejecutar múltiples sistemas operativos y aplicaciones en un solo servidor.
- Acelerar y simplificar la administración de TI, el mantenimiento y la implementación de aplicaciones nuevas.
- Vídeo <http://bcove.me/38mx0561>

# Terminología dentro de la virtualización

- **SO anfitrión (host)** El SO de la PC física donde la VM está corriendo.
- **SO huésped (guest)** El SO que corren en la VM
- **Máquina Virtual (VM)** El ambiente creado para el SO huésped.

# Contenido I

## 1 Control de versiones

- Github
- Git
- Repositorios
- add, commit, push y otros comandos

## 2 Virtualización

- Virtualbox

## 3 Ubuntu

- Requerimientos
- Instalación

## 4 Introducción

- El computador

- Para el curso utilizaremos VirtualBox.
- **Portabilidad** varios SO populares (32 y 64 bits).
  - Hosted Hypervisor (hipervisor tipo 2) por software.
  - VMs pueden exportarse.
- No es necesario virtualización por hardware.
- Buen soporte de hardware (SMP, USB, , red, etc)
- Snapshots (guardar estados de VMs)

# SO host soportados

- Windows (XP 32bits, Server 2003 32bits, Vista, Server 2008, 7, 8 Server 2012 64 bits)
- Mac OS X (10.6, 10.7, 10.8 64 bits, 10.9 64 bits)
- Linux (Debian, Oracle Linux 5 y 6, Redhat, Fedora, Gentoo, openSUSE, Mandriva)
- Solaris 11 y 10 64 bits

# Empezando con VirtualBox

- <https://www.virtualbox.org/wiki/Downloads>
- La instalación es sencilla y fácil de seguir.
- VirtualBox Manager



- Izquierda se listan las VMs
- Los botones permiten crear, configurar y manejar la VM.
- Derecha muestra las propiedades de la VM seleccionada.



# VB manager con varias VMs



# Creando una VM

- Clic “Nueva” y seguir los pasos del asistente.
- **Nombre de la VM** como aparecerá en el manager. Nombre descriptivo “Ubuntu 14.04 64”.
- **Tipo de SO** Windows, **Linux**, Solaris, Mac, etc.
- **Versión** Ubuntu 64 bits
- En la siguiente pantalla: 1024MB de RAM. Está cantidad de memoria no estará disponible en le SO host cuando corramos nuestra VM. Si nos exdemos en la cantidad de memoria nuestro SO se volverá inestable (mucho *swaping*).
- Siguiente, Disco duro virtual para la VM. Este disco puede ser usado en otro host con VB.

# Disco duro virtual



- “Crear un disco duro virtual ahora”
- “VDI”
- “Tamaño fijo”
- 10GB de tamaño
- “Crear”
- Esperar ...

- Seleccionar VM en el administrador
- Ir a “configuración”
- “Almacenamiento”
- Debajo de “Controlador IDE” clic en vacío
- En “Unidad CD/DVD” = IDE secundario maestro.
- Clic en el ícono a la derecha.
- Seleccionar un archivo de disco virtual de CD/DVD y buscar el .ISO de Ubuntu. Aceptar.
- En el administrador seleccionar la VM y clic en “Iniciar” para empezar la instalación de Ubuntu.

# Contenido I

## 1 Control de versiones

- Github
- Git
- Repositorios
- add, commit, push y otros comandos

## 2 Virtualización

- Virtualbox

## 3 Ubuntu

- Requerimientos
- Instalación

## 4 Introducción

- El computador

# Contenido I

## 1 Control de versiones

- Github
- Git
- Repositorios
- add, commit, push y otros comandos

## 2 Virtualización

- Virtualbox

## 3 Ubuntu

- Requerimientos
- Instalación

## 4 Introducción

- El computador

# Requerimientos del sistema

- Procesador 700 MHz (Intel Celeron o mejor)
- 512Mb de RAM
- 6.5GB de espacio en disco
- Resolución de pantalla 1024x768 VGA
- Unidad de CD/DVD o puertos USB para instalación.
- Internet (recomendado)

# Contenido I

## 1 Control de versiones

- Github
- Git
- Repositorios
- add, commit, push y otros comandos

## 2 Virtualización

- Virtualbox

## 3 Ubuntu

- Requerimientos
- Instalación

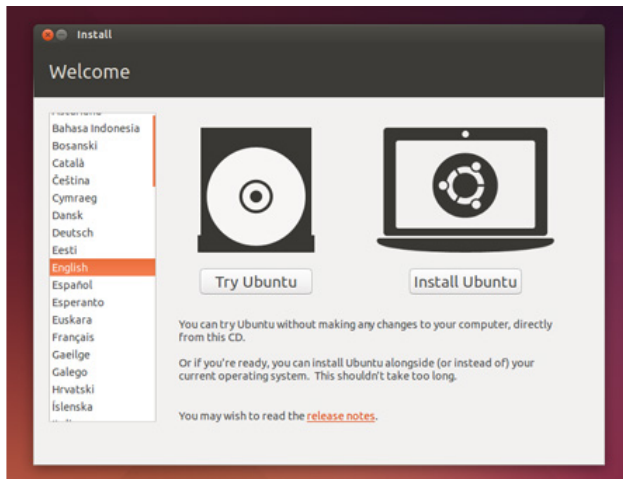
## 4 Introducción

- El computador

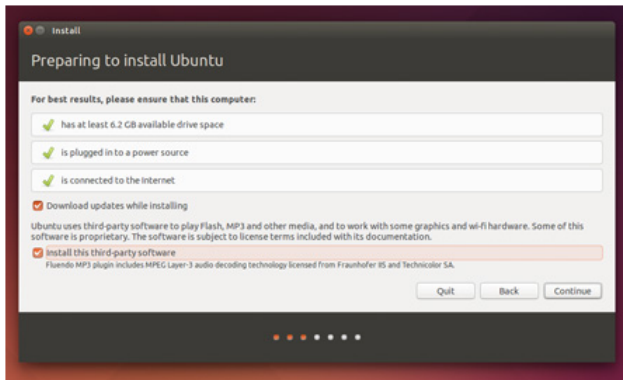


# Instalación

Seleccionar idioma, Instalar Ubuntu.

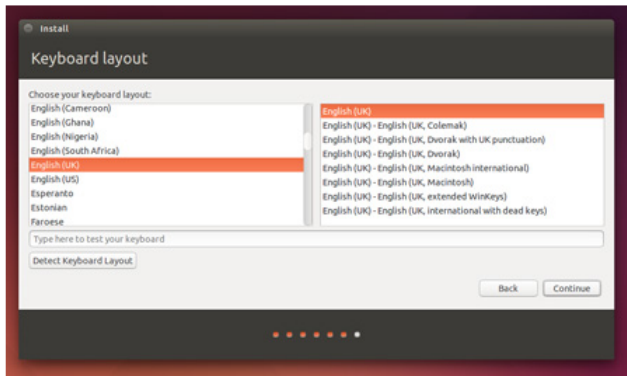


# Chequeo del sistema



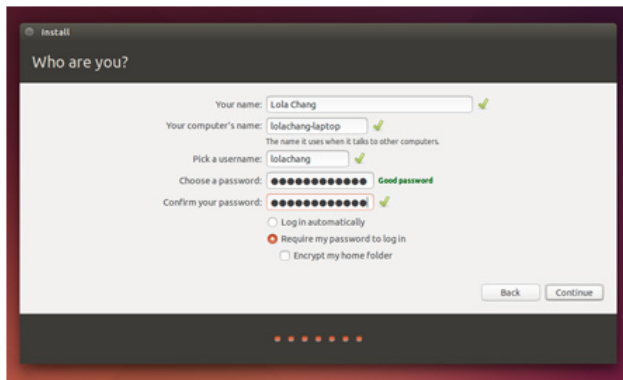
- clic en “Descargar actualizaciones mientras se instala” e “Instalar este software de terceros”. Continuar.
- Seleccionar “Borrar disco e instalar Ubuntu”. Instalar ahora.
- Si sale un mensaje diciendo “Desea escribir los cambios en el disco” clic en continuar.

# Distribución del teclado



- Seleccionar la región. Guayaquil. Continuar.
- Distribución de teclado. Continuar.

# ¿Quién es usted?



The image shows a screenshot of the Ubuntu installer window titled "Install". The main heading is "Who are you?". The form contains the following fields and options:

- Your name: Lola Chang ✓
- Your computer's name: lolachang-laptop ✓  
The name it uses when it talks to other computers.
- Pick a username: lolachang ✓
- Choose a password: [masked] ✓ Good password
- Confirm your password: [masked] ✓
- ☐ Log in automatically
- ☒ Require my password to log in
- ☐ Encrypt my home folder

At the bottom right are "Back" and "Continue" buttons. At the bottom center is a progress bar with 7 dots, the 6th of which is highlighted.

- ¿Quién es usted?
- Esperar...

# Detalles de VirtualBox

- Íconos de la esquina inferior derecha.
- Integración del mouse mientras este esté sobre la ventana de VB.
- Auto captura del teclado cuando la ventana de VB esté activa.
- Tecla host = tecla control derecho
- Pantalla completa

# Contenido I

## 1 Control de versiones

- Github
- Git
- Repositorios
- add, commit, push y otros comandos

## 2 Virtualización

- Virtualbox

## 3 Ubuntu

- Requerimientos
- Instalación

## 4 Introducción

- El computador

# Contenido I

## 1 Control de versiones

- Github
- Git
- Repositorios
- add, commit, push y otros comandos

## 2 Virtualización

- Virtualbox

## 3 Ubuntu

- Requerimientos
- Instalación

## 4 Introducción

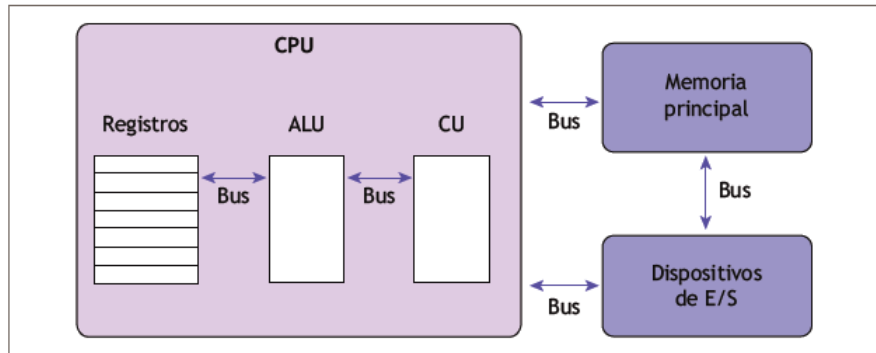
- El computador

# El computador

- El esquema o arquitectura del computador de hoy es muy similar al de los primeros computadores a pesar de su rápida evolución.
- ¿Qué arquitecturas conocen? Harvard, Von Neumann.
- ¿En qué se diferencian las arquitecturas? Programación hardware vs software
- ¿Qué arquitectura utilizan los computadores de hoy? Von Neumann.
- ¿Quién puede graficar la arquitectura de Von Neumann?
- ¿Qué son los registros?



# Arquitectura de Von Neumann



# Arquitectura de Von Neumann

Esta arquitectura tiene tres conceptos clave:

- Los datos e instrucciones se almacenan en una sola memoria de lectura-escritura.
- Los contenidos de esta memoria se direccionan indicando su posición, sin considerar el tipo de dato contenido en la misma.
- La ejecución se produce siguiendo una secuencia de instrucciones (la secuencia puede ser modificada).

- Parte principal del computador, el cerebro.
- Conformado por:
  - Unidad de control
  - ALU
  - Registros
- Recibe datos y envía resultados a través de los buses.
- ¿Qué son los buses? un camino sobre el cuál se transmite información.



# Memoria interna

# Referencias

- <http://www.vmware.com/latam/virtualization/virtualization-basics/what-is-virtualization>
- <https://www.virtualbox.org/manual/ch01.html>
- <https://help.ubuntu.com/community/Installation/SystemRequirements>