**PARTE 1 Recapitulemos Conceptos**

**1. Defina una arquitectura cualquiera de una PC y explique las funciones:**

La arquitectura de una PC típica comprende varios componentes clave, incluyendo la CPU (Unidad Central de Procesamiento), la memoria RAM (Memoria de Acceso Aleatorio), el disco duro (o unidades de estado sólido), la tarjeta madre (o placa base), tarjetas de expansión (como tarjetas gráficas o de red), y dispositivos de entrada y salida (como teclado, ratón y monitor). La CPU es responsable de ejecutar instrucciones y coordinar el funcionamiento de todos los componentes. La memoria RAM almacena temporalmente datos y programas que están en uso. El disco duro almacena permanentemente datos y programas incluso cuando la PC está apagada. La tarjeta madre conecta todos los componentes entre sí y proporciona comunicación entre ellos. Las tarjetas de expansión agregan funcionalidad adicional, como gráficos mejorados o conectividad de red. Los dispositivos de entrada y salida permiten la interacción con la computadora.

**2. Cuál es la mayor diferencia entre las arquitecturas de 32 bits y 64 bits:**

La principal diferencia radica en la cantidad de memoria que pueden direccionar. Los sistemas de 32 bits pueden direccionar hasta 4 GB de memoria RAM, mientras que los sistemas de 64 bits pueden direccionar una cantidad mucho mayor, teóricamente hasta 16 exabytes de memoria.

**3. Qué es un sistema operativo:**

Un sistema operativo es un conjunto de programas y rutinas que coordina el uso del hardware de una computadora y proporciona servicios comunes para programas de aplicación. Actúa como un intermediario entre el hardware de la computadora y el software de la aplicación.

**4. Define Kernel e interfaces de usuario:**

El kernel es la parte central del sistema operativo que gestiona los recursos del sistema, como la memoria, los dispositivos de entrada/salida y el procesamiento del CPU. Las interfaces de usuario son las formas en que los usuarios interactúan con el sistema operativo y los programas, como la interfaz de línea de comandos (CLI) o la interfaz gráfica de usuario (GUI).

**5. Define procesos e hilos:**

Un proceso es una instancia en ejecución de un programa en un sistema operativo, que incluye el código del programa, los datos asociados y el estado del recurso del sistema. Un hilo es una unidad básica de ejecución dentro de un proceso, que puede ejecutar tareas independientes simultáneamente con otros hilos en el mismo proceso.

6**. Dentro de un proceso que son los datos, códigos y pilas, como se relacionan con los hilos del proceso:**

Los datos son variables y estructuras de datos utilizadas por el programa. El código es la secuencia de instrucciones ejecutables del programa. La pila es una región de memoria utilizada para almacenar datos temporales y contexto de ejecución de funciones. Los hilos comparten el mismo espacio de código y datos, pero tienen su propia pila de ejecución.

**7. Poner ejemplo de cómo el sistema operativo organiza los procesos:**

El sistema operativo organiza los procesos utilizando tablas de procesos, que contienen información sobre cada proceso en ejecución, como su identificador único, estado, recursos asignados y prioridad de ejecución como:

**FIFO (First-In, First-Out):**

Este método asigna prioridad a los procesos en función de su tiempo de llegada, es decir, el primero que llega es el primero en ser atendido.

**Round-Robin:**

En este método, cada proceso recibe una pequeña y misma cantidad de tiempo de CPU, llamado cuántum, y luego se pasa al siguiente proceso en la cola aun cuando este no haya terminado. Si un proceso no ha terminado durante su cuántum asignado, se pone al final de la cola para esperar su próximo turno.

**Prioridad dinámica (SJF - Shortest Job First):**

En este método, el proceso con el menor tiempo de ejecución se selecciona para ejecutarse primero. Esto minimiza el tiempo promedio de espera y maximiza la utilización del procesador.

**Prioridad estática:**

Cada proceso recibe una prioridad fija en función de su importancia relativa. Los procesos con mayor prioridad se ejecutan antes que los de menor prioridad. Puede llevar a problemas de inanición si los procesos de alta prioridad monopolizan los recursos.

**8. Poner ejemplo de cómo el sistema operativo maneja la memoria RAM:**

El sistema operativo administra la memoria RAM asignando y liberando espacio para procesos y datos según sea necesario. Utiliza técnicas como la paginación, la segmentación y el swapping para gestionar eficientemente la memoria disponible.

**Paginación:**

* La paginación es una técnica de gestión de memoria que divide la memoria física y virtual en bloques de tamaño fijo llamados "páginas".
* Tanto el espacio de direcciones físicas como virtuales se dividen en páginas del mismo tamaño.
* Cuando un proceso intenta acceder a la memoria, el sistema operativo traduce la dirección virtual del proceso a una dirección física utilizando una tabla de páginas.
* La paginación permite la asignación de memoria de forma flexible y eficiente, ya que los marcos de página pueden asignarse y liberarse individualmente.

**Segmentación:**

* La segmentación es otra técnica de gestión de memoria que divide la memoria en segmentos lógicos de diferentes tamaños.
* Cada segmento representa una región lógica del espacio de direcciones del proceso, como el código, los datos y la pila.
* Los segmentos pueden crecer y contraerse dinámicamente según las necesidades del proceso.
* El sistema operativo utiliza tablas de segmentos para realizar la traducción de direcciones lógicas a direcciones físicas.

**Swapping:**

* El swapping, o intercambio de memoria, es una técnica utilizada cuando la memoria física se agota y es necesario liberar espacio.
* Consiste en mover parte de un proceso de la memoria principal (RAM) a un espacio de almacenamiento secundario, como el disco duro.
* Cuando se necesita nuevamente la parte del proceso que se ha movido al almacenamiento secundario, se vuelve a cargar en la memoria principal.
* El swapping permite que el sistema operativo mantenga más procesos en ejecución que la capacidad física de la memoria, aunque con el costo de una menor velocidad de acceso a los datos que se han movido al disco.

**9. Qué es la memoria virtual:**

La memoria virtual es una técnica que utiliza el sistema operativo para proporcionar a cada proceso la ilusión de disponer de una memoria contigua y privada, independientemente de la cantidad de memoria física real disponible en el sistema. Esta técnica utiliza tanto la memoria RAM (memoria física) como el almacenamiento en disco (memoria secundaria) para administrar la memoria de manera eficiente.

La relación entre la memoria virtual y el swapping radica en cómo se gestiona la memoria cuando la cantidad de memoria física es limitada. Cuando un proceso necesita más memoria de la que está disponible en la RAM, el sistema operativo puede utilizar la memoria virtual para almacenar temporalmente partes menos utilizadas de la memoria en el disco. Este proceso se conoce como "swapping" que explicamos anteriormente.

**10. Qué es el código abierto:**

El código abierto se refiere a software cuyo código fuente está disponible públicamente y puede ser modificado, distribuido y utilizado libremente por cualquier persona.

**11. Poner ejemplo de Windows y Linux respecto al kernel y sus interfaces. Relaciónalo con el código abierto:**

Windows tiene un kernel cerrado y sus interfaces son principalmente gráficas, como el escritorio y las ventanas. Linux, por otro lado, tiene un kernel de código abierto y ofrece una variedad de interfaces, incluyendo la interfaz de línea de comandos y varias interfaces gráficas de usuario como GNOME y KDE.

**12. Que significa LTS:**

LTS significa "Long-Term Support" (Soporte a Largo Plazo) y se refiere a versiones de software que reciben soporte extendido y actualizaciones de seguridad durante un período prolongado, generalmente varios años.

**13. Qué es la virtualización y cuál es su importancia:**

La virtualización es la creación de una versión virtual de un recurso de hardware, como una computadora, un servidor, un sistema de almacenamiento o una red. Permite la ejecución de múltiples sistemas operativos y aplicaciones en una sola máquina física, lo que mejora la eficiencia, la flexibilidad y la utilización de recursos.

**14. Quiénes son el Host y el Guest:**

El host es la máquina física que ejecuta el software de virtualización, mientras que el guest es la máquina virtual que se ejecuta dentro del host.

**15. Qué significa Hipervisor tipo1 y tipo 2:**

Un hipervisor de tipo 1 se ejecuta directamente sobre el hardware físico y gestiona los sistemas operativos invitados, mientras que un hipervisor de tipo 2 se ejecuta como una aplicación dentro de un sistema operativo hospedador y proporciona un entorno para las máquinas virtuales.

**16. Relate los pasos para crear una Máquina Virtual con VirtualBox:**

Los pasos para crear una máquina virtual con VirtualBox incluyen instalar VirtualBox, crear una nueva máquina virtual, asignar recursos como CPU y RAM, configurar el almacenamiento, montar la imagen ISO del sistema operativo invitado, y finalmente arrancar la máquina virtual para instalar el sistema operativo.

**17. Qué significa proporcionar recursos a la máquina virtual:**

Significa asignar recursos físicos como CPU, memoria RAM, almacenamiento y periféricos a la máquina virtual para su uso exclusivo.

**18. Qué es y para qué puede utilizar la clonación y el Snapshot en VirtualBox:**

La clonación en VirtualBox es el proceso de crear una copia idéntica de una máquina virtual existente. Los snapshots son instantáneas del estado actual de una máquina virtual en un momento dado, lo que permite volver a ese estado en caso de problemas o cambios no deseados.

**19. Qué significa el estado NAT en la asignación de red de la máquina virtual:**

NAT (Network Address Translation) es un método que utiliza VirtualBox para asignar direcciones IP a las máquinas virtuales y permitirles acceder a la red externa a través del host, ocultando las direcciones IP internas de las máquinas virtuales.

**20. Es posible Compartir carpetas entre el Host y el Guest:**

Sí, es posible compartir carpetas entre el host y el guest mediante la configuración de carpetas compartidas en la configuración de la máquina virtual.

**21. Diferencia entre ISO y Disco Virtual:**

Un archivo ISO es una imagen de disco que contiene una copia exacta de los datos de un disco óptico, como un CD o un DVD. Un disco virtual, por otro lado, es un archivo que simula un disco duro o una unidad de almacenamiento y puede contener cualquier tipo de datos, incluidos sistemas operativos, aplicaciones y archivos de datos.

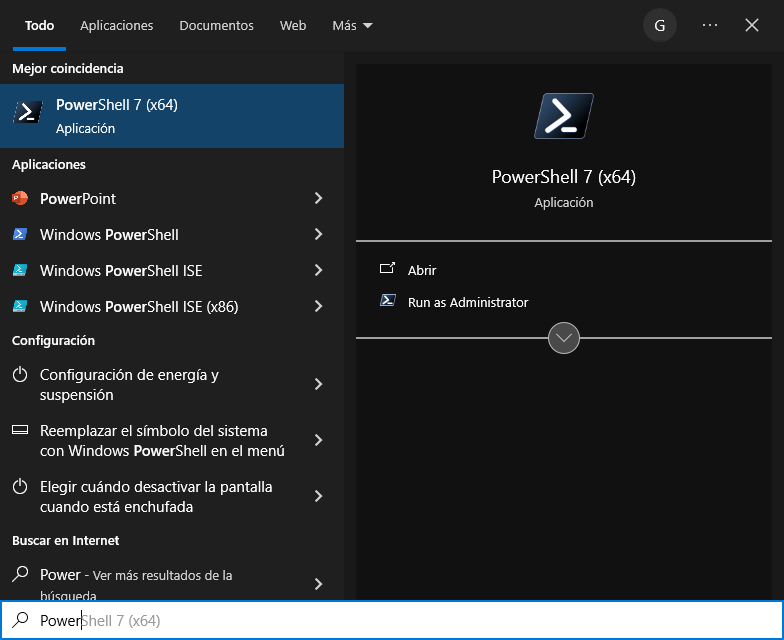
**PARTE 2 Instalando en la práctica: Diversas formas de acceder a diferentes sistemas operativos**

1. **PowerShell**
2. **WSL**
3. **Virtualbox**
4. **Chocolatey**
5. **Vagrant**
6. **Con el uso de GIT**

**1 - PowerShell**

PowerShell es una interfaz de línea de comandos (CLI) y un entorno de scripting desarrollado por Microsoft para la administración de sistemas y la automatización de tareas en sistemas operativos

PowerShell está diseñado para ser más potente que el símbolo del sistema tradicional de Windows (cmd.exe), ya que permite el acceso a una amplia gama de funciones del sistema y la manipulación de objetos y datos de manera más eficiente.



**Los primeros comandos que vamos a estar utilizando**

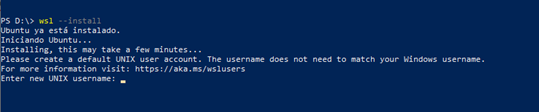
|  |  |  |  |
| --- | --- | --- | --- |
| **Acción** | **CMD (Windows)** | **PowerShell (Windows)** | **Linux/Unix** |
| Listar Carpetas | **dir** | **Get-ChildItem** | **ls** |
| Moverse Entre Carpetas | **cd <nombre\_de\_la\_carpeta>** | **Set-Location <nombre\_de\_la\_carpeta>** | **cd <nombre\_de\_la\_carpeta>** |
| Crear Carpetas | **mkdir <nombre\_de\_la\_carpeta>** | **New-Item -ItemType Directory -Name <nombre\_de\_la\_carpeta>** | **mkdir <nombre\_de\_la\_carpeta>** |
| Eliminar Carpetas | **rmdir <nombre\_de\_la\_carpeta>** | **Remove-Item <nombre\_de\_la\_carpeta>** | **rm -r <nombre\_de\_la\_carpeta>** |

Con el comando Get-Alias dir se puede tener el alias del comando en PowerShell

**2- WSL**

Los desarrolladores pueden acceder a la potencia de Windows y Linux al mismo tiempo en una máquina Windows. Subsistema de Windows para Linux (WSL) permite a los desarrolladores instalar una distribución de Linux (como Ubuntu, OpenSUSE, Kali, Debian, Arch Linux, etc.) y usar aplicaciones, utilidades y herramientas de línea de comandos de Bash directamente en Windows, sin modificar, sin la sobrecarga de una máquina virtual tradicional o una configuración de arranque dual.

**wsl –install**

****

**Solicita usuario y contraseña para Ubuntu**

**Probar los comandos de la tabla**

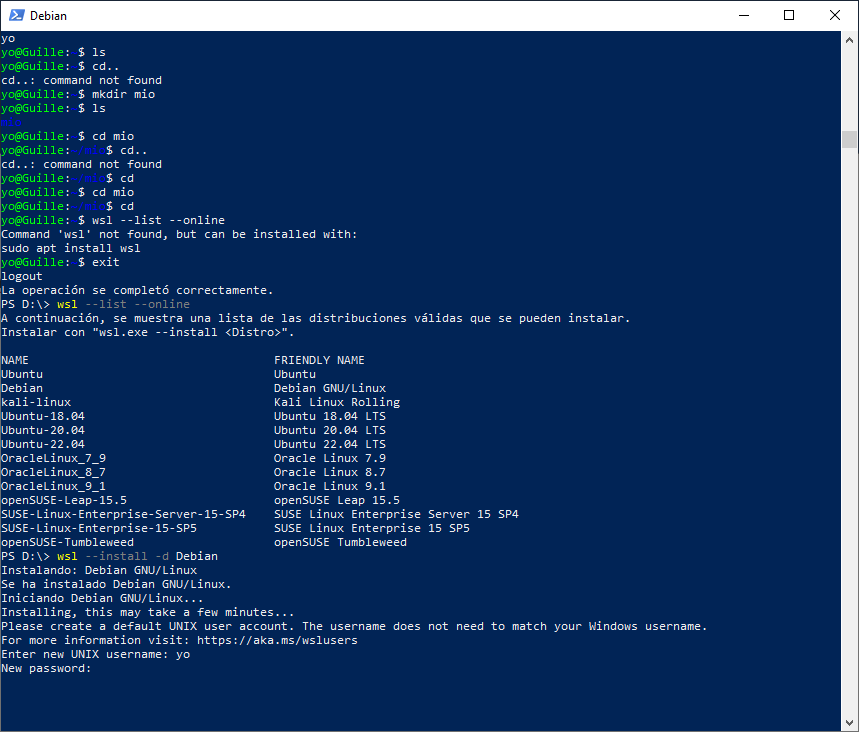
**Podemos Cambiar la distribución predeterminada de Linux instalada**

De forma predeterminada, la distribución de Linux instalada será Ubuntu. Se puede cambiar mediante la marca -d.

Para cambiar la distribución instalada, escriba: wsl --install -d <Distribution Name>. Reemplace <Distribution Name> por el nombre de la distribución que desea instalar.

Para ver una lista de las distribuciones de Linux disponibles para descargar a través de la tienda en línea, escriba wsl --list --online o wsl -l -o.

Para instalar distribuciones de Linux adicionales después de la instalación inicial, también puede usar el comando wsl --install -d <Distribution Name>.



**Instalemos Debian y probemos todos los comandos**

**3- Virtualbox**

Asegurarte de que puedas virtualizar (systeminfo) Ir a la Bios a habilitar

Descargar desde https://www.virtualbox.org/wiki/Downloads

Descargar ISO LTS

Crear y realizar las primeras configuraciones

Asignar recursos

Compartir red

Compartir carpetas entre el host y el guest

Clonar una Máquina

Crear Snapshot

Ir a la terminal y lanzar los primeros comandos

**4 – Chocolatey**

Chocolatey es una herramienta de gestión de paquetes para Windows que simplifica el proceso de instalación, actualización y administración de software en sistemas Windows. Funciona de manera similar a los gestores de paquetes en sistemas operativos basados en Unix, como apt en Debian/Ubuntu o yum en Red Hat/CentOS.

Con Chocolatey, los usuarios pueden instalar rápidamente una amplia variedad de software con solo un comando, en lugar de tener que buscar y descargar manualmente los instaladores individuales de cada aplicación. Además, Chocolatey facilita la gestión de las actualizaciones de software, ya que puede mantener automáticamente los programas instalados actualizados a las últimas versiones disponibles.

Algunas características clave de Chocolatey incluyen:

Fácil instalación de software: Con Chocolatey, puedes instalar software con un simple comando en la línea de comandos.

Administración centralizada de paquetes: Chocolatey proporciona una base de datos centralizada de paquetes de software, lo que facilita la búsqueda y la instalación de programas.

Actualizaciones automáticas: Chocolatey puede mantener tus programas instalados actualizados automáticamente, eliminando la necesidad de buscar manualmente actualizaciones.

Gestión de dependencias: Chocolatey maneja las dependencias entre paquetes de software, lo que garantiza que todas las dependencias necesarias se instalen correctamente.

En resumen, Chocolatey es una herramienta poderosa que simplifica la gestión de software en sistemas Windows, haciéndolo más eficiente y menos propenso a errores. Es especialmente útil para administradores de sistemas y usuarios avanzados que desean automatizar y simplificar el proceso de instalación y mantenimiento de software en sus sistemas Windows.

**Comando para instalar**

**Web:** [**https://chocolatey.org/install**](https://chocolatey.org/install)

**Verificar que estén las políticas activadas en power Shell ejecutar**

PS C:\Windows\system32> Get-ExecutionPolicy

Restricted **(Si sale esto. Ejecutar entonces)**

PS C:\Windows\system32> Set-ExecutionPolicy AllSigned

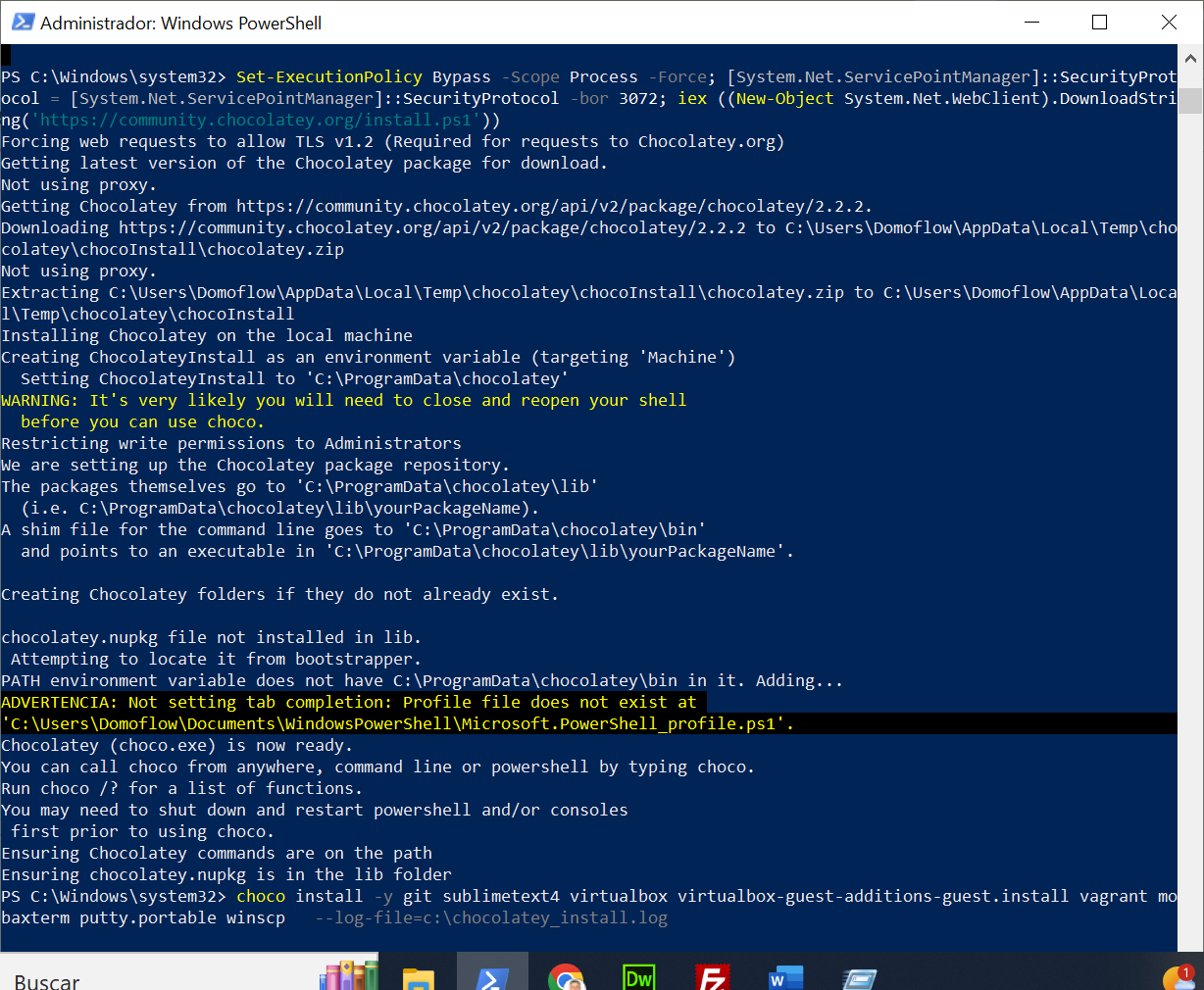
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))

**Posteriormente con este comando se instalan las herramientas vagrant y git que estaremos utilizando desde chocolatey:**

choco install -y git vagrant --log-file=c:\chocolatey\_install.log

**Importante con chocolatey Podemos instalar varias aplicaciones por ejemplo el propio virtual box**

choco install -y virtualbox virtualbox-guest-additions-guest.install --log-file=c:\chocolatey\_install.log



**Vagrant y GIT**

Vagrant es una aplicación libre desarrollada en ruby que nos permite crear y personalizar entornos de desarrollo livianos, reproducibles y portables. Vagrant nos permite automatizar la creación y gestión de máquinas virtuales.

Git es un sistema de control de versiones distribuido, diseñado para rastrear cambios en archivos y coordinar el trabajo entre múltiples personas en proyectos de desarrollo de software.

Descargar el repositorio que está en

<https://github.com/upszot/UTN-FRA_SO_Vagrant>

Utilizar GIT para clonar

Crear en c carpeta

/c/so/repogit

Dentro de repogit dar clic derecho abri bash ejecutar

$ git clone https://github.com/upszot/UTN-FRA\_SO\_Vagrant.git

$ git pull para tener actualizado el directorio

probar los comandos:

# Crear y dejar en running la/s VM definidas en el vagrantfile

vagrant up

vagrant ssh para entrar

# Apagar y Eliminar sin pedir confirmación

vagrant destroy -f

# Apagar la vm sin eliminarla (una forma de mantener persistencia de datos)

vagrant halt

Creemos nuestro primer vagrant file

vagrant init

vagrant init nombre de la distro desde vagrant cloud