

Datos personales:

- **Apellido:** Segnini
- **Nombre:** Martín
- **Legajo:** 50567
- **Comisión:** 04
- **Fecha:** 12/09/2022

1. Son como habilidades personales que se aplican a la hora de trabajar, estas pueden ser parte de nosotros mismos por naturaleza o se pueden ir adquiriendo con la práctica y paso del tiempo. Son muy importantes para el ambiente laboral ya que ayudan al buen trato entre pares, mejores resultados y en general un mejor ambiente de trabajo.
 - a. Trabajo en Equipo: Supongamos que tenemos que entregar un TP en equipo, el buen trabajo en equipo podría ser dividirse las tareas entre compañeros para una mejor organización, ayudar a alguien si tiene alguna duda o problema, no tratar mal a alguien del equipo si está equivocado o tuviste un mal día.
 - b. Facilidad y velocidad a la hora del aprendizaje: Es la capacidad de poder adaptarse a las cosas nuevas, uno de los ejemplos que se dieron en clase había sido que en nuestro trabajo se implementa una nueva tecnología, tener facilidad y velocidad a la hora del aprendizaje sería poder adaptarse y comprender esta tecnología rápidamente.
 - c. Capacidad Resolutiva: Por ejemplo surge algún problema nuevo en nuestro trabajo, tener una buena capacidad resolutiva sería poder pensar con claridad y velocidad una buena solución a este inconveniente sin alterarse.
 - d. Buena gestión del tiempo: Es la capacidad de administrar nuestro tiempo como dice, tenemos que entregar muchos trabajos para la tup, una buena gestión del tiempo haría que no se nos junten todas las fechas y poder entregar cada trabajo en tiempo y forma.
2. Un **Sistema Operativo** podría decirse que es un conjunto de programas que permiten que tanto el Hardware como Software de nuestra PC se comuniquen para poder realizar operaciones como **por ejemplo**, leer los archivos dentro de nuestras unidades de almacenamiento, ejecutar programas, hacer funcionar dispositivos externos como son Mouse, Teclado, Monitor (no sería una característica principal porque algunos tipos de sistemas operativos no necesitan de estos), detectar errores del Hardware o Software y monitorizar los recursos de nuestro equipo entre más funciones. Creo que yo pondría como los **más utilizados a Windows y Android**, seguido de **iOS** ya que hoy en día la gran mayoría de personas tienen una computadora personal (Donde me arriesgo a decir que el 90% usa Windows) o sino un dispositivo móvil donde tienen alguna versión de Android o iOS.

La principal diferencia entre un SO y un Kernel está en que el Kernel vendría

a ser lo que conecta nuestro Sistema Operativo con el Hardware del equipo, se encarga de que nuestro sistema pueda acceder tanto a los recursos físicos como a la información de ellos.

Un **Proceso** sirve para que el sistema operativo entienda cómo ejecutar un programa paso a paso, vendría a ser como el instructivo de cada programa que ejecuta el Sistema

3. La diferencia entre el **Software** y el **Hardware** es que el Software vendrían a ser los programas, procesos y demás información “digital” por así decirlo y el Hardware son todos los componentes principales como secundarios de nuestros equipos.

- a. **Hardware Externo:**

- i. **Monitor:** Sirve para ver lo que estaría saliendo como imagen de nuestra PC
 - ii. **Teclado:** Sirve para mandar info por escrito a nuestro equipo
 - iii. **Joystick:** De igual manera que el teclado pero más simple y generalmente orientado para jugar videojuegos

- b. **Hardware Interno:**

- i. **GPU:** La tarjeta gráfica o Graphics Processing Unit se encarga de todos los procesos o tareas más enfocadas a imagen, vídeo, y renderizado, es un componente bastante interesante porque los más nuevos, enfocados al gaming, son como minicomputadoras con su memoria ram, núcleos y demás microcomponentes que permiten esta calidad tan buena de imagen en modelado 3D como videojuegos
 - ii. **CPU:** El procesador o Central Processing Unit es claramente el componente interno principal de cualquier dispositivo ya que se puede comparar con el cerebro humano, es el encargado de conectar todo, desde los datos como los componentes, sin CPU no tenemos computadora.
 - iii. **Motherboard:** o Placa Madre, Placa Base, es donde conectamos todos los componentes de nuestra PC, cuentan con todo un camino de circuitos y microcomponentes conectados que se encargan de llevar la información de un lado al otro en la PC

4. Yo diría que **Bash** es un programa de muy bajo nivel donde podemos mandar órdenes a nuestro equipo en forma de texto, se puede usar tanto como para crear una carpeta como instalar programas en el equipo o ejecutarlos.

El **Bucle FOR** es un bucle que se ejecuta desde 0 o 1 dependiendo el lenguaje hasta la cantidad determinada por nosotros los programadores, este bucle **se va a ir repitiendo e incrementando o decrementando** hasta la cantidad N de veces que hayamos puesto, un ejemplo podría ser que necesitamos leer una variable “Num” una cantidad de 5 veces, para ahorrarnos leer esta variable 5 veces hacemos un bucle for de 0 a 4 veces que lea Num y listo.

El **condicional IF** como su nombre lo dice, en español “Si...” es un condicional que usamos para ejecutar una acción **sólo si** se cumple una condición determinada, por ejemplo ponemos un número y queremos saber si el número es par, con el condicional if podemos saber esto mismo y mostrar por pantalla un texto “el número es par” sólo si nuestro número es divisible por 2

5. Las **principales diferencias** entre las vm y los contenedores está en la **cantidad de recursos** que se necesitan para cada una, las virtual machine son literalmente computadoras virtuales con sus recursos “físicos” a partir de lo que asignamos con nuestros recursos reales a la hora de crear una en cambio los contenedores son mucho más chicos y no son todo un equipo con su sistema operativo y hardware sino que se queda en ser una aplicación.

Los **contenedores** son más prácticos por ejemplo si una empresa quisiera implementar o desarrollar alguna característica nueva para su app sin poner en riesgo todo el funcionamiento normal de esta. En cambio una **VM** podríamos usarla para probar código o programas mucho más exigentes sin poner en riesgo nuestro equipo, yo personalmente antes del cursado había usado una vm para probar algunos programas que podían llegar a tener un virus, era preferible perder un poco más de tiempo y probarlo ahí a que me infecte mi compu.

Las **imágenes** vendrían a ser como el elemento principal para crear contenedores porque estas vienen como con el contenido principal que después permite ser modificado en los contenedores. Para comunicarnos con nuestro Docker Host tenemos los comandos de “**docker build**” para crear una imagen, “**docker pull**” que sería similar a cuando hacemos un pull en git, traemos la imagen a nuestro equipo y el “**docker run**” que permite ejecutar el contenedor

6. La **problemática** que resuelven los volúmenes es el **espacio** que ocupan los contenedores ya que son como una instancia que permite almacenar datos sin aumentar el espacio de nuestro contenedor y sin depender del estado del mismo.

Kubernetes se utiliza para administrar los contenedores que tenemos de una manera mucho más fácil y rápida, nos comunicamos con estos a partir de los archivos “YAML” o “JSON” y con el comando “kubectl” que es la forma de utilizar los kubernetes.

Las pod son como **contenedores para los contenedores** ya que permiten almacenar **uno o mas** contenedores dentro, el mayor problema es que las pods no pueden interactuar con sí mismas o sea no pueden actualizarse , para lograr esto se usan los replicasets y los deployments que son como plantillas para crear mas del otro... Osea tenemos los Pod que no se pueden modificar a si mismos, para esto surgen los ReplicaSets que generan pods a partir de un pod “padre” y después tenemos los Deployments que generan ReplicaSets a partir de uno “padre” también para así mantener siempre en línea nuestros pods