

Programación para las Ciencias Biológicas

Desarrollar algoritmos con
lenguaje de programación
estructurado de línea de
comandos para
secuencias biológicas

Clase 1



MAESTRÍA EN MAESTRÍA EN BIOLOGÍA
COMPUTACIONAL

La excelencia no se improvisa



1. La *INTRODUCCIÓN*

Bienvenidos al curso de Programación para las ciencias biológicas. Este curso trata del estudio y uso de herramientas computacionales que permiten automatizar de manera confiable los procesos de exploración análisis comparación y manipulación de cadenas de los ácidos nucleicos usando los lenguajes de programación BASH R y Python a través del paradigma pedagógico ignaciano.

¿Qué haremos?

En este curso nos vamos a ayudar de las herramientas computacionales para analizar las cadenas de los ácidos nucleicos las cuales, sin estas herramientas, pueden ser muy complejas y difíciles de estudiar, debido principalmente a la gran extensión de tamaño que estas cadenas pueden tener. Sin embargo, con la ayuda de las herramientas computacionales y los lenguajes de programación BASH, R y Python se pueden hacer análisis de gran envergadura, muchos de los cuales se pueden automatizar y con la ayuda de servidores especializados, los resultados pueden ser obtenidos varias horas después.

¿Por qué lo haremos?

Los procesos de estudio de las cadenas de ADN han evolucionado a través de los años, incorporando las herramientas computacionales como una gran ayuda que permite realizar análisis que hace algunos años hubiera requerido de mucho tiempo o hubieran sido impensables.

Esto se complementa con la relevancia que ha tomado en la actualidad el poder realizar rápidamente análisis de cadenas de ADN o ARN, para poder compararlas entre ellas y estudiar sus mutaciones y variantes. En ese sentido, el hecho de que a poco tiempo de que se identificará el virus SARS-CoV-2, causante del COVID-19, se publicara la cadena completa de su material genético ha permitido que pueda ser analizada por todos los científicos del mundo.

¿Con qué lo haremos?

Con el manejo de las herramientas de programación BASH, R y Python podremos analizar y comparar cadenas de ADN y ARN para encontrar similitudes o diferencias entre ellas, así como también patrones específicos de las cadenas, entre otros.

Bienvenidos a disfrutar de este maravilloso mundo de la computación al servicio de las ciencias biológicas

2. Cada asignatura deberá estar dividida en **8 (ocho) clases**, conteniendo:

Contenido:

<p>Cada clase deberá mostrar temas y subtemas contenidas en 8 páginas y un rango de 2800 a 3000 palabras, de autoría del contenidista contratado estar de acuerdo con el syllabus de la asignatura y la distribución de temas y subtemas determinados en el Formulario de Estructuración de la Asignatura (Formulario 1). En este contenido puede agregar gráficos e imágenes debidamente referenciados de acuerdo a la normativa APA versión 7.</p>	<p>Cumple</p>
<p>En cada clase, deberán presentarse, como mínimo, 2 (dos) enlaces relacionados con los temas y subtemas complementando al texto redactado. Para cada enlace deberá haber una breve descripción de lo que se está abordando, presentada de forma dialogada en el contenido. Estas descripciones y enlaces deben estar destacados en verde. Los enlaces pueden ser a:</p>	
<p>- Videos [Se incluyen anuncios, documentales, fragmentos de películas, entrevistas u otros tipos que puedan ser utilizados como material didáctico complementario en la disciplina].</p>	<p>https://youtu.be/btNcfYViubs</p> <p>https://seracademia.com/academia/Bio_Info/M1/Videos/Instalacion-Debian12-server-Jupyter.mp4</p> <p>https://seracademia.com/academia/Bio_Info/M1/Videos/Cuaderno0_BASH.mp4</p>
<p>- Sitios web [Enlaces de sitios que contengan artículos, entrevistas, investigaciones u otro material que pueda ser utilizado como material didáctico complementario en la disciplina].</p>	<p>Cumple</p>
<p>- Otros [Pueden ser archivos adjuntos por el responsable del contenido y enviados junto con el contenido de la disciplina].</p>	<p>Cuderno jupyter: Cuaderno0_BASH.ipynb</p>

3. Al final de cada clase deberán presentarse, obligatoriamente, todas las **REFERENCIAS CITADAS** en el texto del contenido, de acuerdo con la normativa de la American Psychological Association (APA) versión 7 o con la información mínima para ser adecuadas a las normas. Las citas autor-fecha utilizadas a lo largo del contenido deben estar destacadas en azul.

4. Dentro del contenido de la clase subraye **al menos 2 términos** de los temas abordados y, al final coloque las definiciones correspondientes.

5. **RECURSOS DE PROFUNDIZACIÓN:** Son recursos de creación propia del contenidista, aquí podrá explicar conceptos y definiciones que requieran una **explicación más detallada** usando videos e infografías, tablas, estudios de caso, resolución de actividades, presentación de resolución de situaciones reales u otros recursos que ayuden a la comprensión del contenido. Se requiere la entrega de 1 a 3 recursos por clase.

DESARROLLO

1. INTRODUCCIÓN DE LA CLASE

En esta primera clase prepararemos nuestro ecosistema para usar Estructuras de Programación en Línea de Comandos (BASH) optimizado para Biología Computacional utilizando una máquina virtual con la distribución Debian 12 que incluye Jupyter Notebook, configurado con los kernels BASH, R y Python, lo que permitirá ejecutar y desarrollar scripts en estos lenguajes de forma interactiva. En primer lugar, es importante tener en cuenta que Debian 12 está basada en la versión estándar de Linux, con un gran número de paquetes disponibles para instalar a través del sistema de paquetes APT (Advanced Package Tool). También ofrece una excelente estabilidad y seguridad, lo que permite trabajar sin preocupaciones sobre la integridad de los datos.

En el siguiente paso, veremos cómo importar y usar las máquinas virtuales tanto con entorno gráfico, así como la versión servidor con Jupyter Notebook con los kernels BASH, R y Python. Estos kernels permiten ejecutar y depurar scripts escritos en estos lenguajes de forma interactiva dentro del entorno de Jupyter. Además, a discreción del estudiante se tiene la posibilidad de instalar otros kernels como Julia® para ampliar aún más las posibilidades del ecosistema.

Clase 1:

Desarrollar algoritmos con lenguaje de programación estructurado de línea de comandos para secuencias biológicas.

1. Estructuras de programación en lenguaje de comandos (BASH)

1.1. Preparación del ecosistema bioinformático

1.1.1. Lububio como sistema operativo ligero para bioinformática

1.1.1.1. Introducción a Lubuntu como entorno eficiente para tareas bioinformáticas.

1.1.1.2. Instalación y configuración de herramientas necesarias para manejo de secuencias biológicas.

1.1.2. Configuración de la Máquina Virtual “Servidor Debian 12” con Jupyter y entornos de desarrollo BASH, R y PYTHON

1.1.2.1. Instalación y configuración de Debian 12 como servidor bioinformático. *Configuración de Jupyter Notebooks en un servidor Debian para el desarrollo de scripts de análisis de secuencias biológicas en BASH.

1.1.2.2. Instalación de paquetes esenciales para bioinformática en línea de comandos.

1.2. Scripts en línea de comandos (BASH)

1.2.1. Scripting en BASH, variables, control de flujo y estructuras de decisión.

1.2.2. Comparación de BASH con otros lenguajes de scripting usados en bioinformática.

Referencias citadas en la Clase 1.

Videos:

1. Instalación y configuración de Lubuntu con herramientas para el manejo de secuencias biológicas.

<https://youtu.be/btNcfYViubs>

2. Uso y familiarización con Lubuntu

https://seracademia.com/academia/Bio_Info/M1/P1/Presentacion_Entorno.html#/instalacin-de-la-imagen-iso-de-lubuntu-r-en-la-mquina-virtual.

3. Configuración de la Máquina Virtual “Servidor Debian 12” con Jupyter y entornos de desarrollo BASH, R y PYTHON.

https://seracademia.com/academia/Bio_Info/M1/Videos/Instalacion-Debian12-server-Jupyter.mp4

4. Instalación de paquetes esenciales para bioinformática en línea de comandos y variables en BASH.

https://seracademia.com/academia/Bio_Info/M1/Videos/Cuaderno0_BASH.mp4

Cuadernos de trabajo Jupyter

Cuaderno0_BASH.ipynb

Referencias Bibliográficas

1. Biolinux® (Biolinux for Scientists, 2014). Disponible en: <https://www.bioinformatics.org/groups/?group_id=341>
2. Lubuntu 20.04 (Lubuntu Development Team, 2020). Disponible en: <<https://lubuntu.me/focal-released/>>
3. LXDE (LXDE Project, 2024). Disponible en: <<https://www.lxde.org/>>
4. Oracle VirtualBox® (Oracle Corporation, 2024). Disponible en: <<https://www.virtualbox.org/>>
5. GNU Bash Project (Free Software Foundation, 2024). Disponible en: <<https://www.gnu.org/software/bash/>>
6. Python (Python Software Foundation, 2024). Disponible en: <<https://www.python.org/>>
7. The R Project for Statistical Computing (R Development Core Team, 2024). Disponible en: <<https://www.r-project.org/>>

Definición de los términos citados en la Clase 1.

1. BASH: BASH es un acrónimo que significa “Bourne Again Shell” y es un intérprete de comandos (shell) que fue desarrollado como una mejora del original Bourne Shell (sh), creado por Stephen Bourne. BASH es una de las shells más populares en sistemas operativos basados en Unix, como Linux y macOS y es el intérprete de comandos de línea de código abierto utilizado en la mayoría de las distribuciones de Linux y Unix.

2. Oracle Virtualbox: Es un emulador de máquinas virtuales de código abierto desarrollado por Oracle Corporation que permite crear varias máquinas virtuales dentro de una única máquina física, lo que facilita la prueba y despliegue de diferentes sistemas operativos y aplicaciones.

3. LXDE: Es un entorno de escritorio ligero y rápido diseñado para computadoras con recursos limitados o sistemas operativos lentos.

4. Biolinux: Es una distribución de Linux específica para la bioinformática que incluye software y herramientas populares utilizadas en este campo, como Biopython y Bedtools.

5. Máquina anfitriona: Es la computadora o servidor en el que se ejecuta una o varias máquinas virtuales. También es conocida como host.

6. Máquina huésped: Es una máquina virtual creada dentro de una máquina anfitriona que puede ejecutar un sistema operativo diferente al del anfitrión, lo que permite instalar y utilizar software o aplicaciones específicas sin interferir con el sistema operativo original. También es conocida como cliente o invitado.

7. Lubuntu: Es una distribución de Linux basada en Ubuntu que utiliza el entorno de escritorio LXDE (Lightweight X11 Desktop Environment).

8. R: Es un entorno de análisis estadístico y gráficos basado en el lenguaje de programación. Es muy popular entre los científicos, investigadores y estadísticos debido a su potencia y flexibilidad.

9. Python: Es un lenguaje de programación de alto nivel diseñado para ser fácil de aprender y utilizar en todas las tareas. Es muy popular entre los desarrolladores debido a su sintaxis clara, legibilidad y facilidad para la creación de scripts e interfaces gráficas.

Profundización Clase 1.

Clase 1:

Desarrollar algoritmos con lenguaje de programación estructurado de línea de comandos para secuencias biológicas.

1. Estructuras de programación en lenguaje de comandos (BASH)

En bioinformática BASH es muy útil para analizar, estudiar, comparar y extraer patrones de cadenas de ADN o ARN y realizar conversiones entre ellas, de tal manera que se facilita el análisis de grandes volúmenes de datos bioinformáticos. Adicionalmente, al desarrollar los scripts de automatización se ahorra tiempo volviendo los procesos de análisis más eficientes o incorporando la posibilidad de calendarizar procesos cuyo tiempo de ejecución, dado el tamaño de los datos, pudiera ser muy extenso limitándose el investigador a ingresar las instrucciones y esperar el resultado durante varias horas o días.

1.1. Preparación del ecosistema bioinformático

Para este curso se requiere de un entorno que tenga los siguientes elementos:

- Terminal BASH
- Entorno de programación para Python ®
- Entorno de programación para R ®

De manera ilustrativa, en caso de que el estudiante no haya tenido experiencias previas con el sistema operativo GNU/Linux®, se han preparado dos máquinas virtuales para facilitar la instalación y configuración de las herramientas necesarias para este curso.

La primera máquina virtual incluye un ecosistema con entorno gráfico denominado Lububio.

La segunda máquina virtual, en cambio, está pensada como un ecosistema de un servidor, el cual no cuenta con entorno gráfico. Sin embargo, el estudiante puede acceder a este recurso a través de un navegador web desde su equipo anfitrión, utilizando el servicio Jupyter® Notebook, que tiene habilitado por defecto los kernel BASH, R® y PYTHON®. A través de las libretas de trabajo en Jupyter®, el estudiante podrá desarrollar y replicar todas las actividades planificadas para este curso.

1.1.1. Lububio como sistema operativo ligero para bioinformática

Lububio está basado en la distribución liviana Lubuntu® y tiene las herramientas necesarias que le permitirán al estudiante conocer de este sistema operativo e interactuar nativamente con su entorno gráfico derivado del cliente de escritorio LXDE®.

1.1.1.1. Introducción a Lubuntu como entorno eficiente para tareas bioinformáticas

Lububio tiene preinstalado un ecosistema que permite realizar tareas dentro del campo de la biología computacional usando el entorno gráfico LXDE® de Lubuntu®. Cuenta con las siguientes herramientas:

- Bash
- Spyder3
- Rstudio

Dado que el objetivo de esta máquina virtual es desarrollar o fortalecer la experiencia en el uso del entorno gráfico liviano LXDE del sistema operativo GNU/Linux®, se recomienda al estudiante seguir las instrucciones de importación y uso básico de esta máquina virtual.

1.1.1.2. Instalación y configuración de herramientas necesarias para manejo de secuencias biológicas.

Revise este video [<https://youtu.be/btNcfYViubs>], y realice las siguientes actividades:

Para importar la imagen OVA, primero descárguela desde este sitio [<https://seracademia.com/academia/Lububio.ova>], una vez que la haya descargado, desde Oracle VirtualBox® impórtela como un servicio virtualizado, como se muestra en la Figura 1.



Figura 1. Importación del servicio virtualizado Lububio

En la siguiente ventana ubique la imagen Lububio.ova donde la haya guardado en su disco duro, como se presenta en la Figura 2.

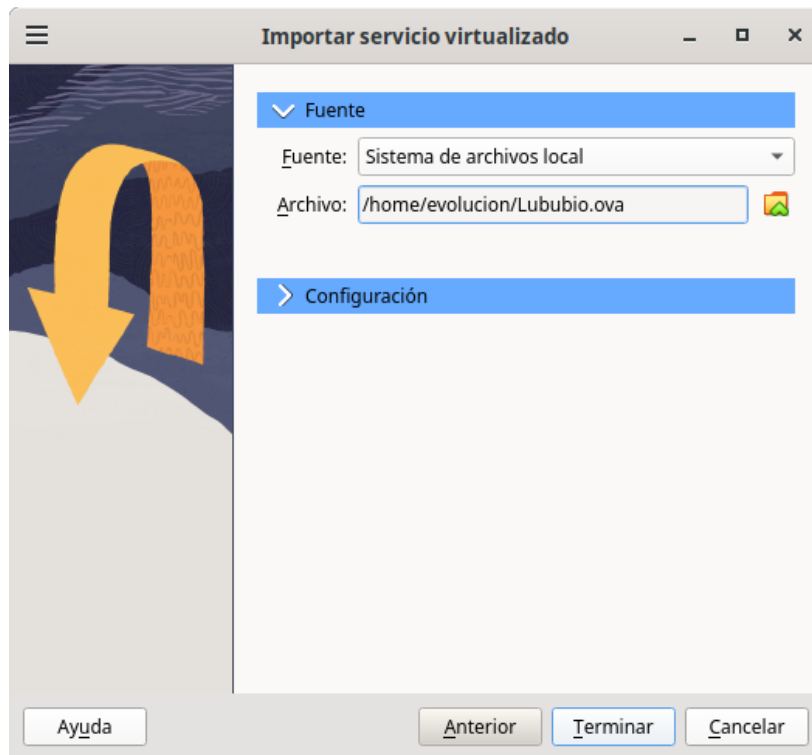


Figura 2. Localización de la imagen Lububio.ova

· Acepte el acuerdo de licencia, como se muestra en la Figura 3.

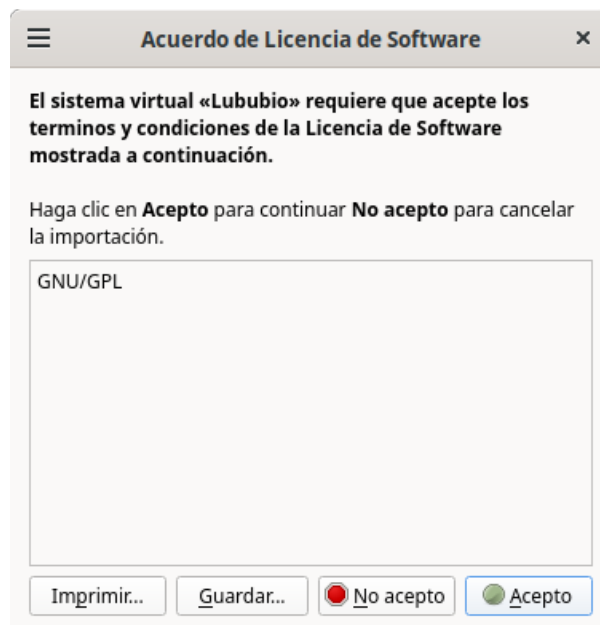


Figura 3. Acuerdo de licencia de Software



Espera a que termine el proceso de importación, como se indica en la Figura 4.

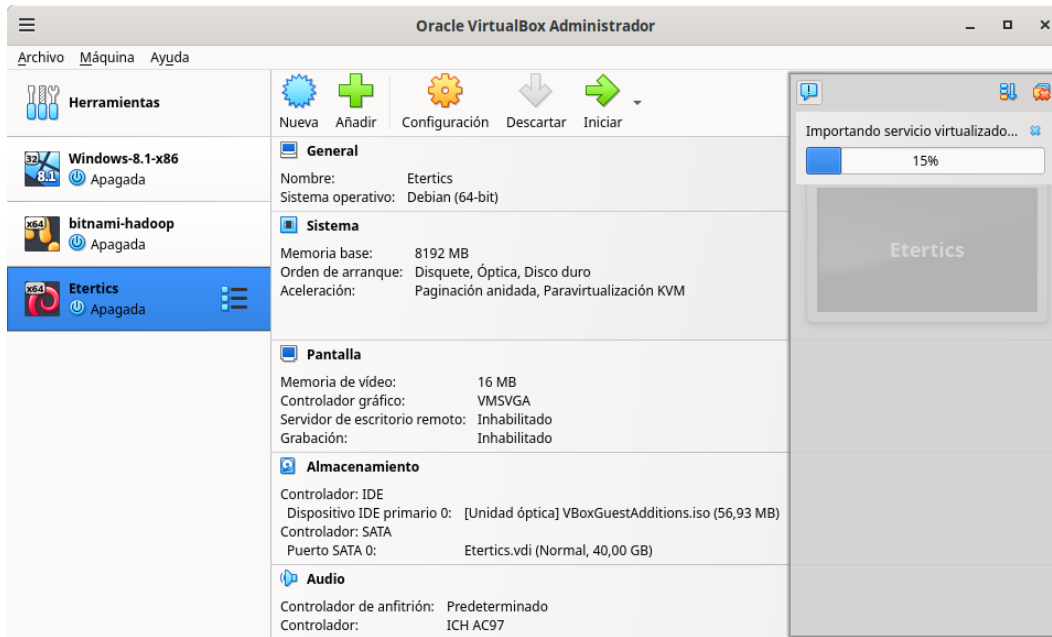


Figura 4. Avance de la importación del servicio virtualizado Lububio

El tiempo dependerá de la velocidad y capacidad de su computadora (máquina anfitriona). Cuando el proceso termine deberá tener una nueva máquina virtual llamada Lububio, como se presenta en la Figura 5.

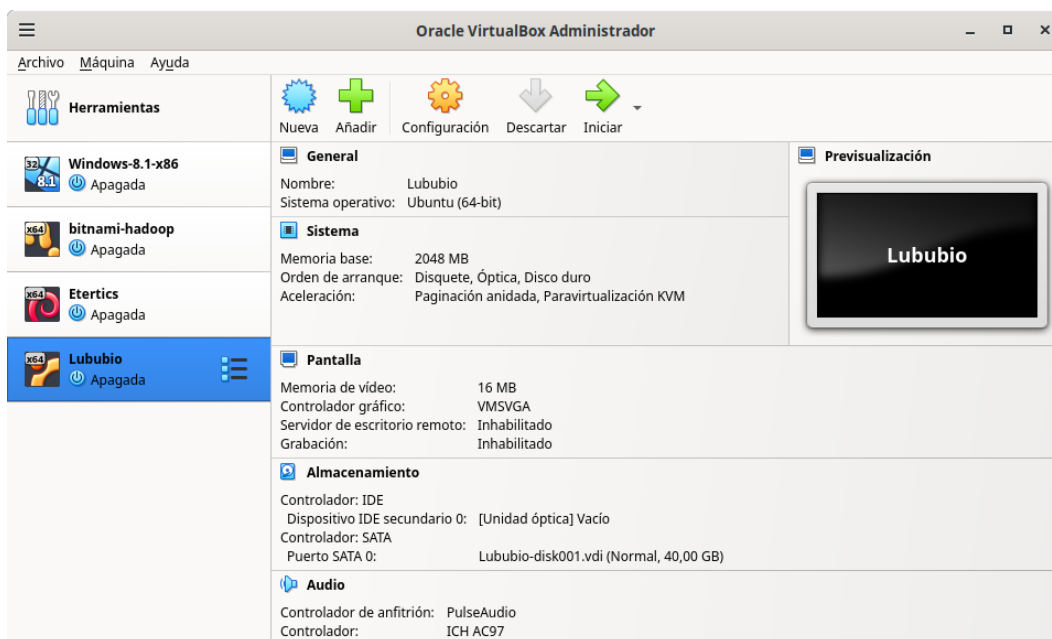


Figura 5. Máquina virtual Lububio

El siguiente video puede servirle de referencia en el proceso de importación:

<https://www.youtube.com/watch?v=Mzhn34Fx70o>

Arranque y Menú de Inicio

- Para arrancar la máquina, hágalo desde Oracle VirtualBox®
- Una vez que se haya iniciado, busque el menú de inicio en la esquina inferior izquierda.
- Pruebe abrir aplicaciones en los 4 escritorios de trabajo.
- Identifique los conceptos “Máquina Anfitriona” y “Máquina Virtual”
- Interactúe con el botón mostrar el escritorio.

Trabajando con la máquina virtual

- Identifique los accesos directos a los tres entornos de trabajo:
 - Terminal BASH
 - IDE Spyder para Python
 - IDE Rstudio para R
- Identifique en el menú de inicio como arrancar los tres entornos de trabajo.
- Identifique las coincidencias de esta máquina con el viejo Bio-Linux:
 - Acceda a la documentación de referencia de Bio-Linux
 - Identifique que paquetes se podrían instalar en nuestra máquina virtual (**sudo apt install nombre_paquete**)
 - Revise la descripción de algunos de los paquetes de la documentación de referencia de Bio-Linux
 - Revise el material introductorio y guía de usuario de Bio-Linux como referencia para nuestra máquina virtual.
- Interactúe con la interfaz gráfica:
 - Use el gestor de archivos

- Identifique su carpeta de usuario “manager” y sus subcarpetas
- Identifique el escritorio
- Identifique la papelera
- Recuerde que muchos programas funcionan desde el terminal sin necesidad de un entorno gráfico.
- Interactúe con la interfaz gráfica:
 - Cree un archivo en el Escritorio
 - Borre el archivo creado
 - vacíe la papelera
- Explore el sistema operativo y abra los programas desde el menú de inicio.
- Abra el terminal QTerminal e interactúe con el:
 - Lance el navegador Firefox desde el terminal
 - Cierre el navegador Firefox interrumpiendo su ejecución
 - Familiarícese con el terminal y las rutas del sistema de ficheros
- Pruebe los comandos propuestos en el video.
- Use el terminal QTerminal e interactúe con el:
 - Determine la ruta en la en que se encuentre
 - Identifique los tipos de ficheros que se pueden listar en QTerminal
 - Conozca como regresar de manera sencilla a su carpeta de usuario
 - Aprenda a usar el atajo de teclado
 - Aprenda a navegar a través del árbol del sistema de ficheros.
- Vea el siguiente video y pruebe los comandos propuestos en el:

<https://www.youtube.com/watch?v=btNcfYViubs>

1.1.2. Configuración de la Máquina Virtual “Servidor Debian 12” con Jupyter y entornos de desarrollo BASH, R y PYTHON

1.1.2.1. Instalación y configuración de Debian 12 como servidor bioinformático.

Revise este video [https://seracademia.com/academia/Bio_Info/M1/Videos/Instalacion-Debian12-server-Jupyter.mp4] , y realice las siguientes actividades:

- Para importar la imagen OVA, primero descárguela desde este sitio [<https://seracademia.com/academia/Debian12-server-v2.ova>], una vez que la haya descargado, desde Oracle VirtualBox® impórtela como un servicio virtualizado, como se muestra en la Figura 6.

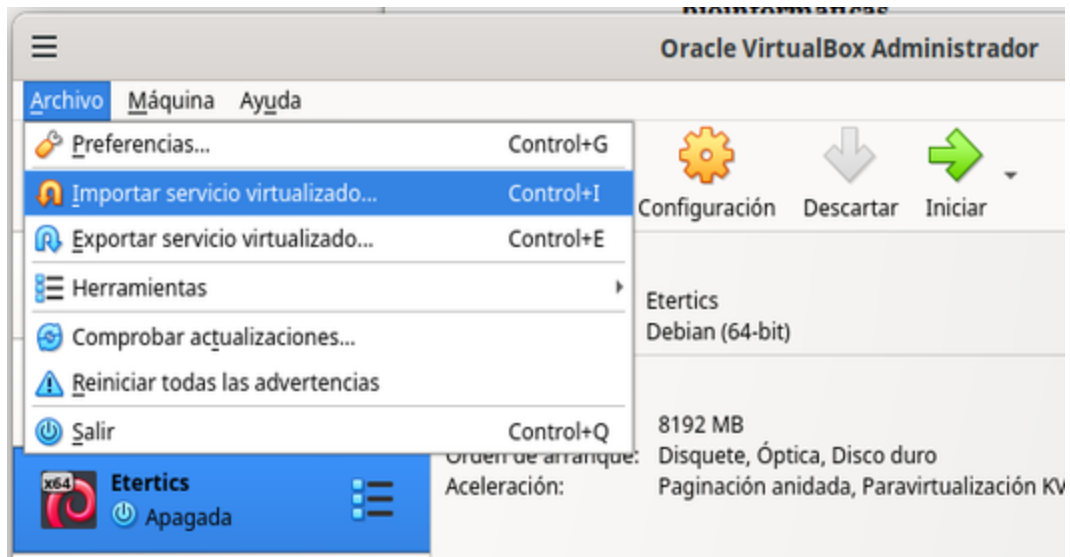


Figura 6. Importación de la imagen ova Debian 12

- En la siguiente ventana ubique la imagen Debian12-server-v2.ova donde la haya guardado en su disco duro, como se indica en la Figura 7.

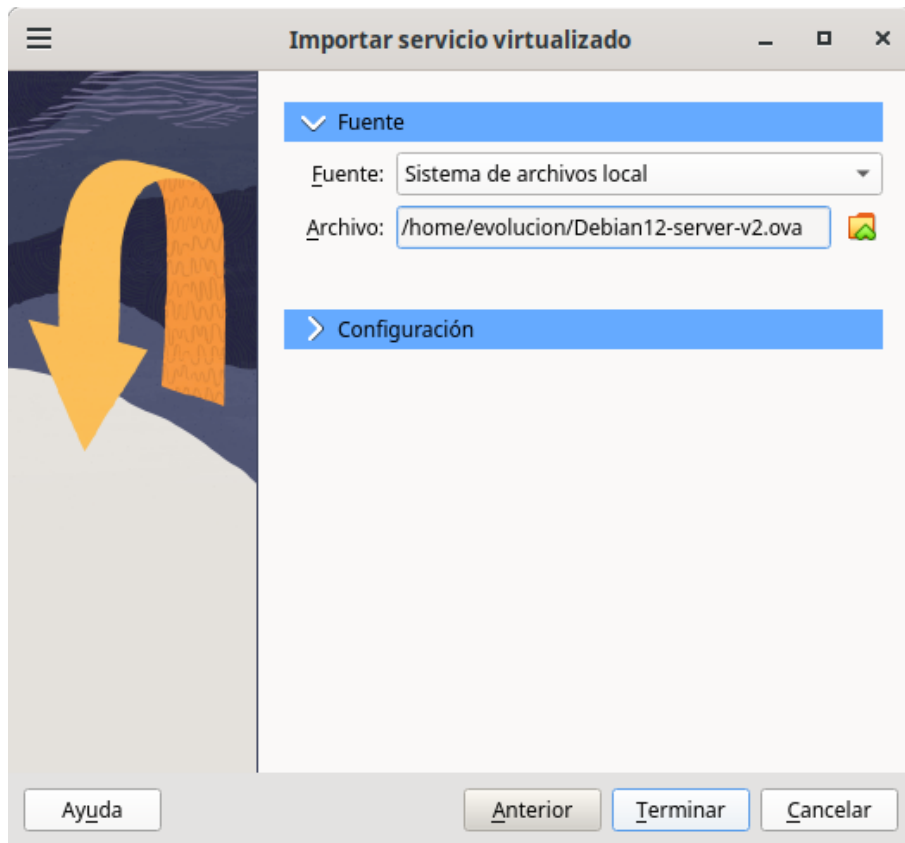


Figura 7. Localización de la imagen Debian 12

Espera a que termine el proceso de importación, como se indica en la Figura 8.

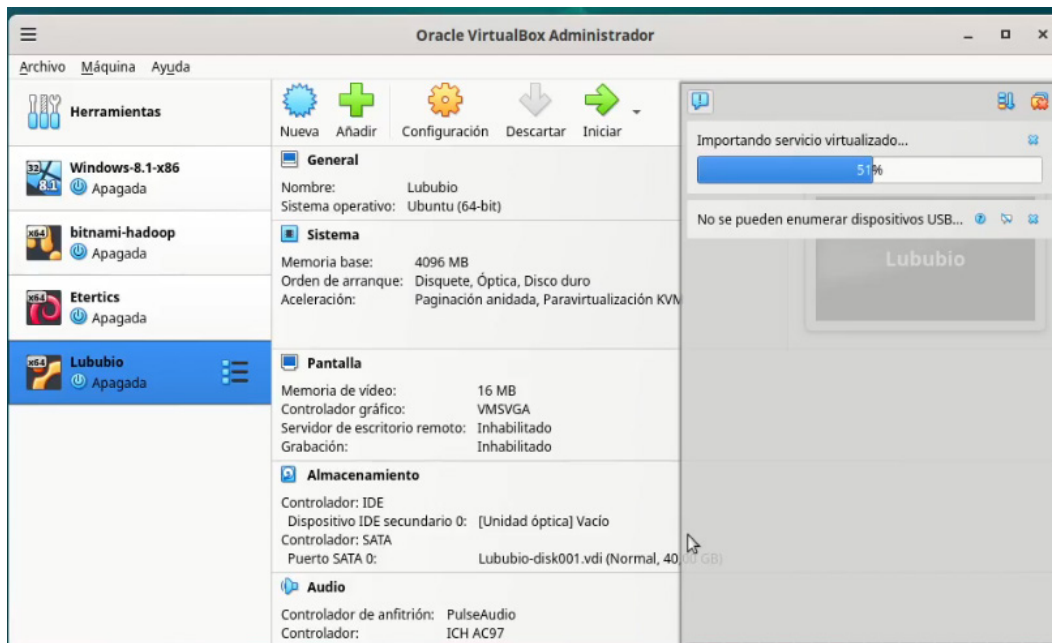


Figura 8. Avance de la importación del servicio virtualizado Debian 12

Cuando el proceso termine deberá tener una nueva máquina virtual llamada Debian12-server, como se muestra en la Figura 9.

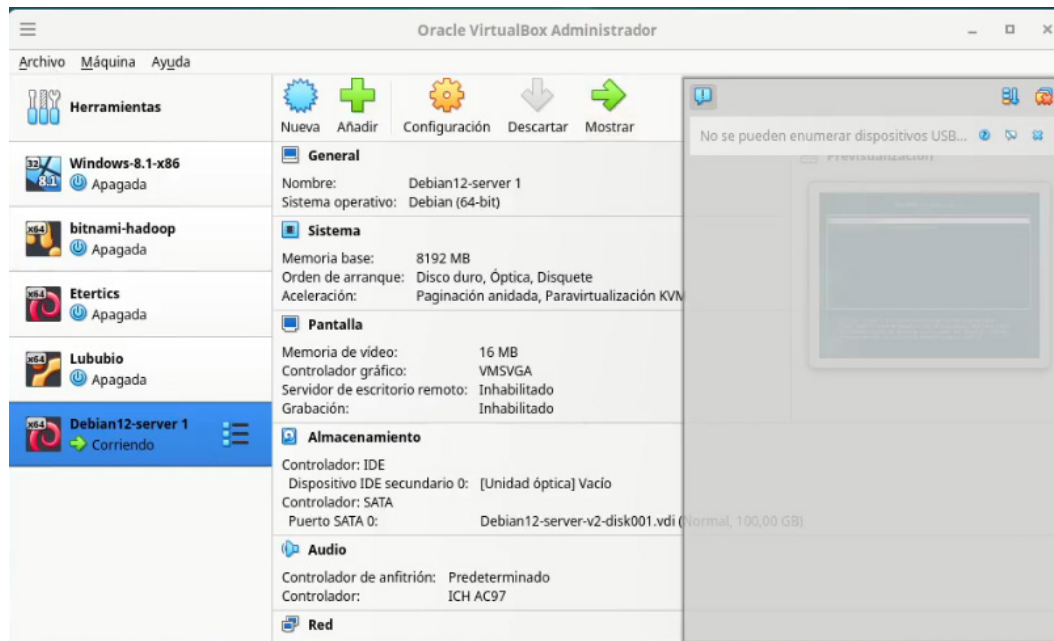


Figura 10. Máquina virtual Debian 12

1.1.2.2. Configuración de Jupyter Notebooks en un servidor Debian para el desarrollo de scripts de análisis de secuencias biológicas en BASH

En la máquina virtual con el ecosistema de Debian 12 como servidor al que se le ha retirado el entorno gráfico para ganar en eficiencia y velocidad, se tiene Jupyter previamente configurado con el kernel BASH. Esto permite ejecutar comandos y utilizar herramientas para bioinformática y análisis de secuencias directamente en el ambiente de trabajo de Jupyter a través de un navegador web de la máquina anfitriona.

Este entorno activa la instalación previa del lenguaje de script BASH y proporciona acceso a una variedad de herramientas populares para el análisis de bioinformática, tales como Biopython, Bedtools, Samtools, Pysam, Fastq, y otras herramientas útiles para investigadores y científicos que trabajan en bioinformática ya que proporciona un ambiente completo de análisis sin la necesidad de que el estudiante tenga que instalar herramientas individuales o configurar el sistema operativo.

También permite a los usuarios utilizar comandos en línea y scripts BASH directamente des-

de el terminal SH y desde este llamar a BASH, o desde la interfaz de Jupyter Notebook para facilitar el trabajo de análisis de bioinformática y la reproducibilidad de los resultados.

Adicionalmente, en este entorno de trabajo de Jupyter Notebook con el kernel BASH en Debian 12, no solo podemos utilizar las herramientas populares para bioinformática mencionadas anteriormente, sino también podemos ejecutar los comandos unix como `tail`, `grep`, `sed` y `awk`. Esto es muy útil en el análisis de datos de bioinformática ya que permite realizar tareas de procesamiento de archivos, filtrado y manipulación de datos rápidamente y de manera eficiente.

Por ejemplo, podemos utilizar `tail` para ver las últimos N líneas de un archivo en un Jupyter Notebook:

1.1.2.3. Instalación de paquetes esenciales para bioinformática en línea de comandos

Para instalar paquetes esenciales para bioinformática en GNU/Linux a través de línea de comandos, se lo puede hacer de dos maneras. La primera, cuando se tiene los programas binarios diseñados específicamente para la distribución, y en el caso de Debian se lo puede hacer usando el gestor de paquetes APT (Advanced Package Tool) que viene preinstalado en la distribución o en su defecto. En caso de que no se cuente con los binarios de la distribución, la segunda manera es a través de compilar los paquetes desde las fuentes del programa que se quiere instalar.

El administrador de paquetes **apt**, es un utilitario que gestiona los archivos binarios de software compilado y sus dependencias. Hay diferentes administradores de paquetes disponibles para cada distribución de este sistema operativo como, por ejemplo:

1. **apt** (Advanced Package Tool) para sistemas basados en Debian (Ubuntu, Kali Linux, etc.)
2. **yum** (Yellowdog Updater, Modified) para sistemas basados en Red Hat (CentOS, Fedora, RHEL, etc.)
3. **dnf** (Dandified Yum) que es una actualización de yum con mejoras en performance y funcionalidades
4. **pacman** para Arch Linux
5. **apk** para Alpine Linux

El proceso de instalación de paquetes con estos administradores de paquetes generalmente sigue un proceso similar:

1. Abrir una terminal y acceder a la cuenta de usuario con privilegios de root o sudo.
2. Buscar el paquete deseado utilizando el administrador de paquetes (por ejemplo, apt-cache search nombre_paquete).
3. Instalar el paquete utilizando el comando específico del administrador de paquetes.
4. Verificar que el paquete se ha instalado correctamente y que está disponible en la carpeta binaria predeterminada (usualmente /usr/bin o /usr/local/bin).

En el cuaderno de trabajo “**cuaderno0_BASH.ipynb**” disponible en los recursos de la plataforma se desarrollan las actividades que se mencionan a continuación.

Para ilustrar el proceso de instalación se instalarán dos paquetes populares para bioinformática, el primero a través de APT y el segundo compilando el paquete. Como bono y de manera inicial en el cuaderno de trabajo se instalará el paquete **cowsay**:

```
sudo apt install cowsay
```

1. Librería Biopython:

Dado que para la librería Biopython se cuenta con los binarios para la distribución Debian®, esta se encuentra disponible en sus repositorios, lo que facilita su instalación con el comando **apt**.

La biblioteca Biopython es software abierto y multipropósito, escrita en Python, destinada a facilitar el análisis y la manipulación de datos genéticos y bioinformáticos. Biopython proporciona una variedad de utilidades para manipular archivos genéticos comunes (como FASTA, GenBank, SwissProt), así como herramientas para análisis secuencias de ADN, ARN y proteínas, análisis estructural, visualización gráfica y más. También incluye funciones para procesar datos de experimentación, integrar datos externos (como los proporcionados por la base de datos NCBI) y realizar análisis estadísticos y bioinformáticos avanzados.

Para instalar Biopython, ejecute el siguiente comando:

```
sudo apt-get install python3-biopython
```

2. Bedtools:

Para instalar un paquete desde las fuentes en GNU/Linux utilizando `make` y `make install`, primero se necesita descargar los archivos fuente del repositorio `git` u obtenerlos a partir de un repositorio del desarrollador de la aplicación. Una vez que tenga las fuentes del programa puede seguir los siguientes pasos:

1. Descargar las fuentes del repositorio del desarrollador, por ejemplo, `git`. Para ello debería tener instalado **git**, en caso de no tenerlo instálelo con **`apt-get install git`**.
2. Ubíquese dentro de la carpeta que descargó o descomprimió y revise la documentación del desarrollador.
3. Instale las dependencias requeridas para compilar el paquete, estas dependencias deberían constar en la documentación del programa. Esto puede ser realizado utilizando `apt-get`. Dependiendo del sistema operativo. En algunos casos, también es posible instalar las dependencias directamente desde las fuentes si las mismas están disponibles en su repositorio.
4. Configurar la instalación utilizando el comando **`make`** o **`configure`**, dependiendo de cómo lo haya establecido el desarrollador. En algunos casos, esta opción puede estar automatizada mediante una llamada a un archivo `setup.sh` o `configure.sh`. Esto configura el paquete para trabajar con el sistema operativo y las dependencias correctas. Para determinar la opción correcta revise detenidamente la documentación del programa.

Como ejemplo ilustrativo se instalará `bedtools`

Para instalar `Bedtools` en su última versión desde sus fuentes, podemos utilizar `Git` para descargar la última versión, instalar las dependencias necesarias y luego ejecutar los scripts de instalación:

Dependencias:

Actualice los repositorios de dependencias

```
sudo apt update
```

Instale las dependencias:

```
sudo apt install build-essential
```

```
sudo apt install liblzma-dev
```

```
sudo apt install liblz-dev  
sudo apt install libbz2-dev  
sudo apt install git
```

Descargar el paquete con GIT:

```
git clone https://github.com/arq5x/bedtools2.git
```

Compilar e instalar:

```
cd bedtools2  
make  
sudo make install
```

1.2. Scripts en línea de comandos (BASH)

Los Scripts en Línea de Comandos (BASH) son una herramienta esencial para la Bioinformática. Estos scripts permiten automatizar tareas repetitivas y personalizar el flujo de trabajo según las necesidades.

En ese sentido se pueden crear scripts para todo tipo de tareas, desde la extracción de datos hasta la visualización de resultados, lo que ayuda a aumentar la productividad y reducir errores humanos.

Adicionalmente, es importante tener en cuenta que los scripts de BASH son muy fáciles de aprender y utilizar, con solo un poco de práctica, se pueden crear scripts eficientes y efectivos que ayuden a simplificar el trabajo en Bioinformática.

Mas adelante en este curso elaborará sus propios scripts con BASH.

1.2.1. Scripting en BASH, variables, control de flujo y estructuras de decisión.

Conceptualmente; scripting en BASH o BASH scripting incluye la creación de scripts que permitan automatizar tareas repetitivas y personalizar el flujo de trabajo según las necesidades, en ese sentido se pueden utilizar variables, control de flujo y estructuras de decisión.

BASH scripting se verá con mayor detalle más adelante. Por ahora y de manera introductoria es importante conocer cómo funcionan y que tipo de variables tenemos en BASH.

Variables

Las variables en BASH son utilizadas para almacenar valores que pueden ser reutilizados dentro del script. Se pueden crear variable asignándoles valores indistintamente de si son números o cadenas de caracteres, ya que en este lenguaje no se requiere declarar previamente las variables y estas toman la naturaleza de lo que se les asigne.

```
variable1="cadena de caracteres"
```

```
variable2=3
```

El control de flujo permite dirigir el flujo de ejecución de un script según las necesidades, así se puede utilizar condicionales y ciclos para lograrlo. Por ejemplo:

```
if [ $variable == "valor" ]  
then  
    echo "La variable tiene el valor deseado."  
fi  
````
```

Las estructuras de decisión en los scripts permiten tomar diferentes acciones según lo que ocurra durante la ejecución del script, para ello se puede utilizar case, switch o select. Por ejemplo:

```
case $variable in
 "valor1")
 echo "Se ha seleccionado valor 1."
 ;;
 "valor2")
 echo "Se ha seleccionado valor 2."
 ;;
esac
```

### 1.2.2. Comparación de BASH con otros lenguajes de scripting usados en bioinformática

#### 1. Características Generales:


- BASH es un intérprete de comandos de línea de código abierto que se utiliza en Linux y Unix.
- R es un lenguaje de programación estadístico orientado a la ciencia de datos.
- Python es un lenguaje de programación multipropósito, utilizado tanto para ciencia de datos como para desarrollo web.

#### 2. Uso en Bioinformática:

- BASH es muy popular en bioinformática debido a su sencillez y flexibilidad. Es fácil de aprender y se puede utilizar para automatizar tareas repetitivas.
- R también es muy utilizado en bioinformática, especialmente para análisis estadísticos. Se trata de un lenguaje muy potente y adaptable que permite trabajar con datos biológicos.
- Python también se utiliza en bioinformática, tanto para el análisis de datos como para la creación de herramientas y paquetes. Es muy flexible y ofrece una gran cantidad de bibliotecas y paquetes para diferentes tareas.

#### 3. Ejemplo de uso:

- BASH puede utilizarse para automatizar tareas repetitivas, como la extracción de datos o la visualización de resultados. Por ejemplo, un script BASH podría extraer secuencias de ADN de un archivo BAM y crear un archivo BED.
- R se utiliza especialmente para análisis estadísticos en bioinformática. Por ejemplo, un script



R podría analizar datos de expresión génica y crear gráficas que muestren los resultados.

- Python también se utiliza para análisis estadísticos en bioinformática, pero también se puede utilizar para la creación de herramientas y paquetes. Por ejemplo, un script Python podría analizar datos de expresión génica y crear un paquete que permita automatizar el proceso de análisis.

De manera general; BASH es un shell de comandos utilizado en sistemas operativos Unix / Linux que permite a los usuarios ejecutar comandos, redirigir la entrada/salida y crear scripts para automatizar tareas repetitivas. Python es un lenguaje de programación multipropósito utilizado para una variedad de aplicaciones como análisis de datos, Machine Learning, robótica y mucho más. R es un lenguaje estadístico y computacional utilizado principalmente para análisis estadístico y científico.

En sistemas operativos GNU/Linux BASH es una buena opción para ejecutar comandos o automatizar tareas repetitivas, mientras que de manera complementaria Python o R son las mejores opciones para analizar datos o realizar cálculos estadísticos.

Tabla 1: Comparación entre BASH, Python y R

| Característica                                      | BASH                                                  | Python                                                        | R                                                                   |
|-----------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------|---------------------------------------------------------------------|
| Propósito principal                                 | Shell scripting, automatización de tareas             | Desarrollo de software, automatización, ciencia de datos      | Análisis estadístico y visualización de datos                       |
| Facilidad de aprendizaje                            | Moderada, basada en comandos de texto                 | Alta, sintaxis clara y legible                                | Moderada, especialmente para estadística                            |
| Popularidad                                         | Alta en administración de sistemas y DevOps           | Alta en desarrollo general y ciencia de datos                 | Alta en estadística y análisis de datos                             |
| Sintaxis                                            | Sencilla pero precisa, basada en comandos             | Fácil de leer y escribir, más estructurada                    | Especializada en estadísticas y matrices                            |
| Ámbito de uso                                       | Administración de sistemas, scripts de automatización | Desarrollo de aplicaciones, análisis de datos, automatización | Análisis de datos, modelado estadístico                             |
| Bibliotecas y paquetes                              | Limitado en comparación con Python y R                | Gran cantidad de bibliotecas (como NumPy, pandas, matplotlib) | Gran cantidad de paquetes para análisis estadístico                 |
| Interactividad                                      | Limitado a la línea de comandos                       | Interactivo en consolas y notebooks (Jupyter)                 | Interactivo en consolas y notebooks (RStudio)                       |
| Velocidad de ejecución                              | Muy rápida para tareas simples de administración      | Moderada, depende de la tarea                                 | Moderada, optimizada para estadísticas                              |
| Soporte para programación orientada a objetos (OOP) | No soporta OOP                                        | Soporta OOP completamente                                     | Limitado soporte para OOP                                           |
| Compatibilidad multiplataforma                      | Sí (Linux, macOS, Windows)                            | Sí (Linux, macOS, Windows)                                    | Sí (Linux, macOS, Windows)                                          |
| Comunidad y documentación                           | Amplia comunidad en administración de sistemas        | Gran comunidad de desarrolladores y científicos de datos      | Gran comunidad en estadísticas y ciencia de datos                   |
| Entornos de desarrollo                              | Terminal o editores de texto                          | IDEs como PyCharm, VSCode, Jupyter Notebooks                  | RStudio, Jupyter Notebooks                                          |
| Velocidad de desarrollo                             | Rápido para tareas simples, menos flexible            | Rápido y flexible para desarrollo general                     | Rápido en tareas estadísticas, menos flexible en desarrollo general |

## Resumen

- Hemos preparado nuestro ecosistema de trabajo importando tanto la máquina virtual con entorno gráfico que permitirá a los estudiantes interactuar con el entorno gráfico LXDE® de GNU/Linux® y la distribución Ubuntu® así como nuestro servidor Debian 12 con Jupyter y los kernel BASH R y Python.
- También hemos visto como arrancar nuestras máquinas virtuales, para usar nuestro entorno de trabajo, ya sea a través del escritorio del sistema operativo como desde Jupyter y un navegador desde la máquina anfitriona.
- Instalamos nuevos paquetes en nuestro sistema operativo tanto con apt como compilándolo y vimos nociones básicas de BASH scripting y los tipos de variables de BASH.
- Finalmente, comparamos BASH con Python y R, lenguajes de programación usados ampliamente en biología computacional y que también veremos en este curso.



**La excelencia no se improvisa**

síguenos

