

UNIVERSIDAD TECNOLÓGICA DEL NORTE DE GUANAJUATO

Área Académica:
Tecnologías de la Información

Programa Educativo:
Redes Inteligentes y Ciberseguridad

Asignatura:
Automatización de Infraestructura Digital I

Unidad I:
Entornos de desarrollo en automatización de redes

Grupo: GRIC4091

Alumno:	N. de control
Venado Soria German Emiliano	1221100894

Docente:
Gabriel Barrón Rodríguez

INTRODUCCIÓN

La automatización de la infraestructura consiste en usar tecnologías que ejecutan tareas con muy poca intervención humana para controlar el hardware, el software, los elementos de red, el sistema operativo (SO) y el almacenamiento de datos, los cuales se utilizan para ofrecer servicios y soluciones de tecnología de la información. (RedHat, 2019)

Las herramientas de automatización de red admiten funciones que van desde la asignación básica de redes y el descubrimiento de dispositivos a flujos de trabajo más complejos, como la administración de la configuración de red y el aprovisionamiento de recursos de red virtual.

La automatización de red también desempeña un papel clave en las redes definidas por software, la virtualización de redes y la orquestación de redes, lo que permite el aprovisionamiento automático de inquilinos y funciones de redes virtuales, como el equilibrio de carga virtual. (Jessica Scarpati, 2018)

A lo largo de este instrumento se podrá encontrar las herramientas empleadas para llevar a cabo la automatización de res descubriendo su contexto, su proceso de instalación y pruebas de funcionalidad de las mismas.

Índice

DESARROLLO	1
Herramientas de Desarrollo	1
Linux Mint	1
Procedimiento de instalación	1
Python	6
Visual Studio Code	6
Procedimiento de instalación	7
Git	8
Procedimiento de instalación	8
Docker	10
Procedimiento de instalación	10
Ejecución	12
Angular	13
Procedimiento de instalación	13
Ejecución	14
Alternativas de mejora en la automatización de procesos y servicios tecnológicos	17
Metodología de desarrollo ágil	17
Historias de usuario	18
Backlog	21
Sprint Planning	21
Comunidades de colaboración, desarrollo y automatización de redes	22
Mapa mental	22
CONCLUSIÓN	23
BIBLIOGRAFÍA	24

DESARROLLO

Herramientas de Desarrollo

Linux Mint

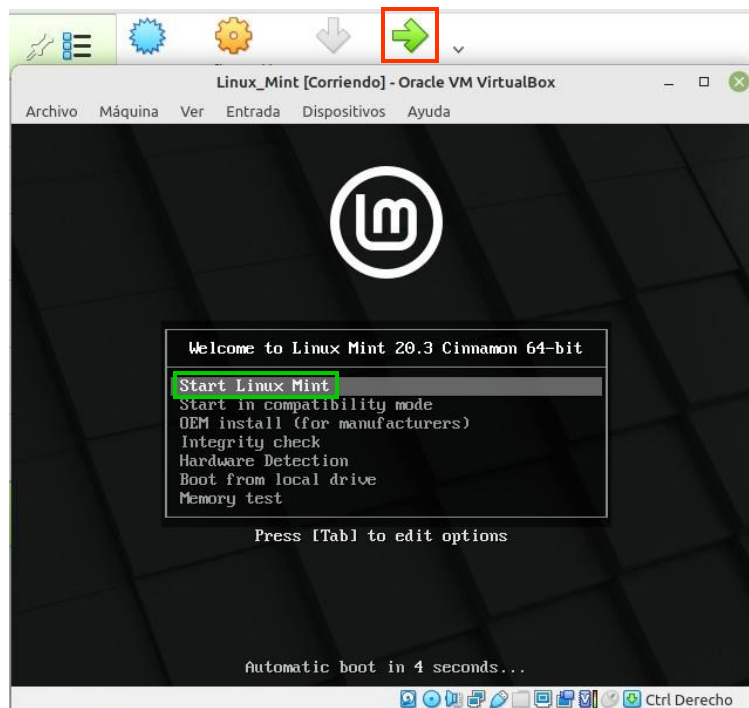
Se trata de una distribución de Linux que tiene como objetivo ofrecer un sistema operativo fácil de usar, elegante y potente. Linux Mint está mayoritariamente dirigido a usuarios de ordenadores de escritorio y ofrece una compilación de aplicaciones pensada para un amplio rango de usos, donde destacan las funciones multimedia y las ofimáticas. (Fernando García, 2022)

El sistema operativo Linux ha resultado muy favorable para desarrolladores, programadores y administradores de servidores, ya que cuenta con herramientas de programación que permiten crear todo tipo de softwares y un entorno gráfico amigable e intuitivo. (Valio, 2022)

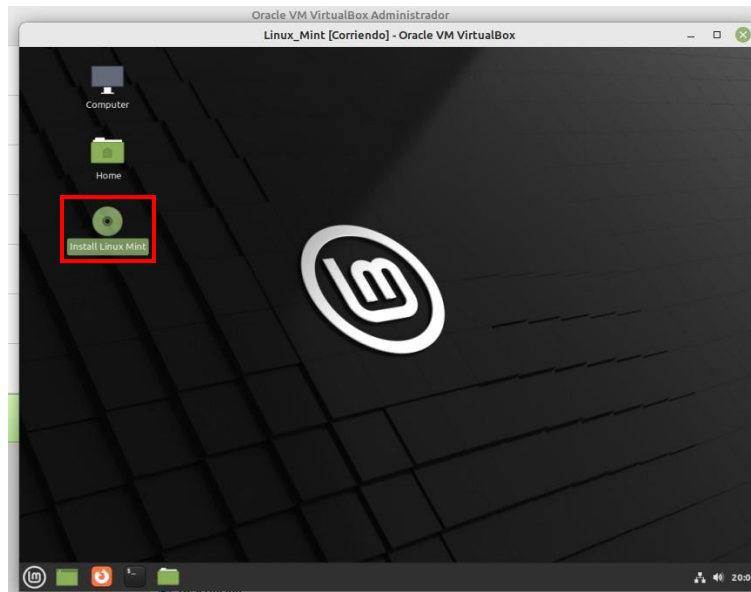
Procedimiento de instalación

El proceso de instalación de Linux Mint Cinnamon es de una manera sencilla, puesto a que se lleva a cabo de manera gráfica. Hay distintas maneras para instalar el sistema operativo (SO), ya sea de manera nativa con la opción Dual Boot, esto se hace cuando deseas tener más de un SO en tu ordenador, o a manera de práctica con una máquina virtual (MV). En este procedimiento se partirá de una MV creada en VirtualBox, para agilizar y capturar claramente el proceso, por lo que, si desconoces los pasos para crear y ejecutar una MV, en los siguientes vínculos encontrarás los pasos y para crear una VM y también los requisitos para la instalación Dual.

1) Lo primero será ejecutar la MV en el icono de “Iniciar” y seleccionar la primera opción del arranque:



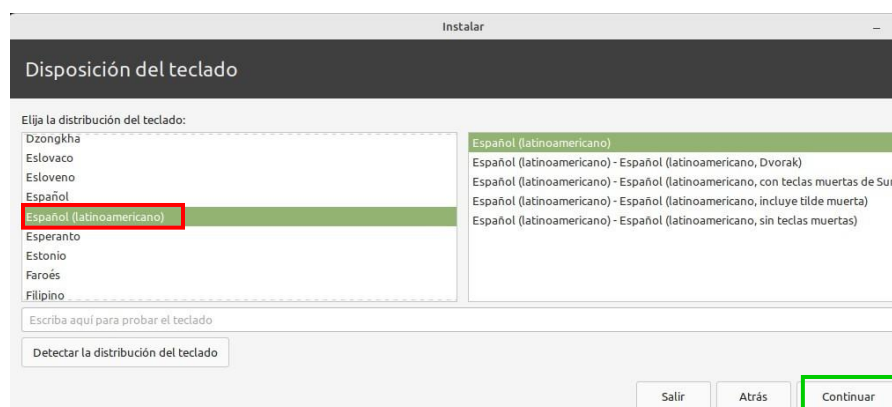
2) Ahora en este escritorio seleccionaremos la opción de “Instalar Linux Mint”:



3) Como primer paso de instalación, debemos seleccionar el idioma que deseamos emplear en el SO, en esta ocasión se opta por el idioma “Español” y clic en “Continuar”:



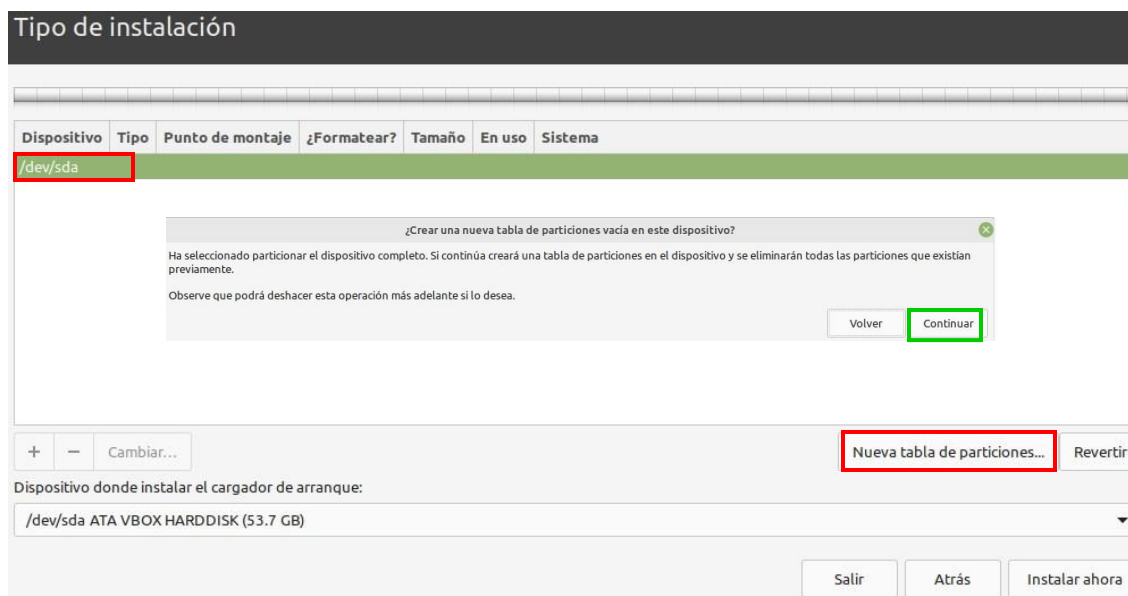
4) Ahora prosigue a seleccionar el idioma del teclado, que será Español Latinoamericano:



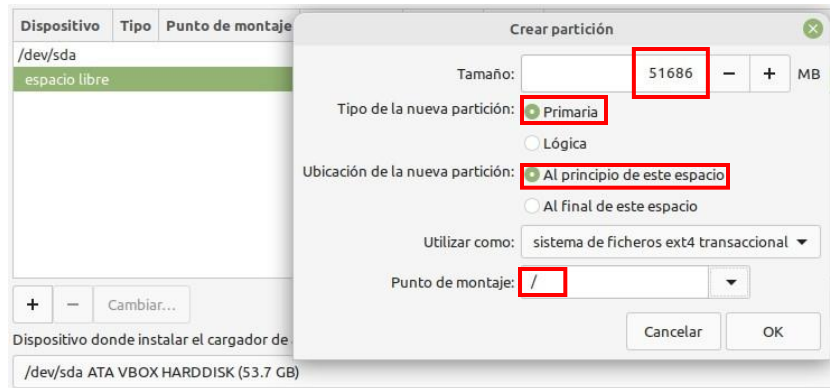
5) En el tipo de instalación, para novatos se recomienda la primera opción de “Borrar disco e Instalar”, pero en esta ocasión se muestra la instalación con la segunda sugerencia de “Más opciones”:



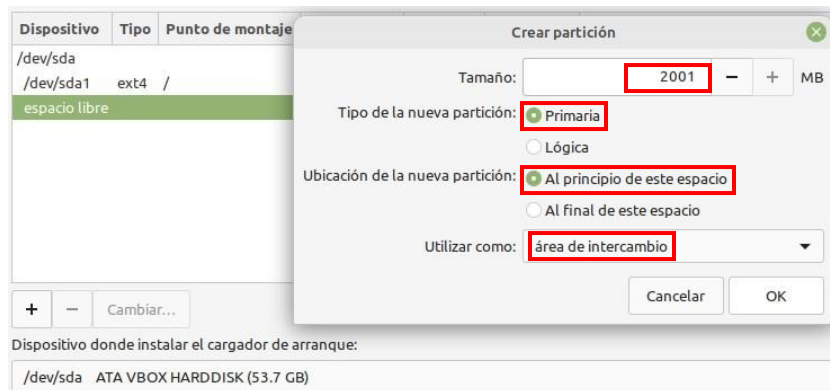
6) Con esta opción podrás crear las particiones necesarias para tu SO Linux Mint. Selecciona la unidad que aparece en pantalla “/dev/sda” y procede a seleccionar “Nueva tabla de particiones” y acepte la creación de la nueva tabla de particiones:



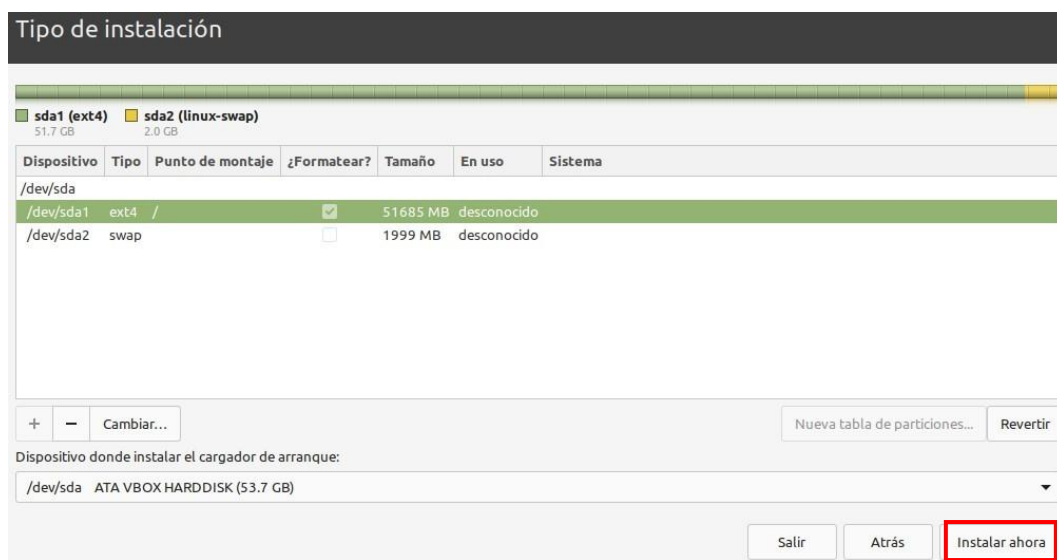
7) Al tamaño brindado debemos aplicar una resta de 2000 MB (2GB) para poder asignar a una segunda partición. Y esta primera será asignada como “Primaria” y “Al principio de este espacio” al montaje raíz (boot ó /):



8) Una MV consumirá memoria RAM de la que le hemos asignado, y es muy limitada, por lo que la única partición que podría ayudar es la partición SWAP, que emulará RAM en disco. Deberá ser el tamaño restante (2000MB) como partición “Primaria” y “Al principio de este espacio” utilizada como “área de intercambio”:



9) Aquí se muestran las dos particiones que acabamos de crear y comenzamos a “Instalar ahora”:



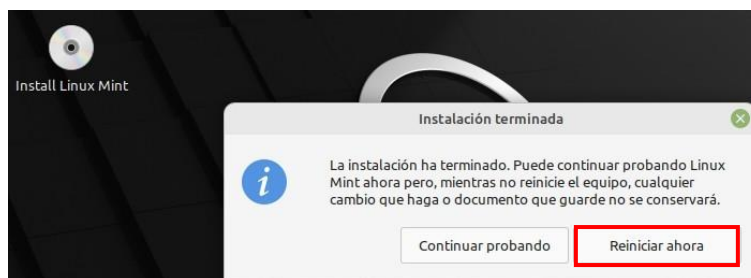
10) Debemos elegir la ubicación donde nos encontramos (México City):



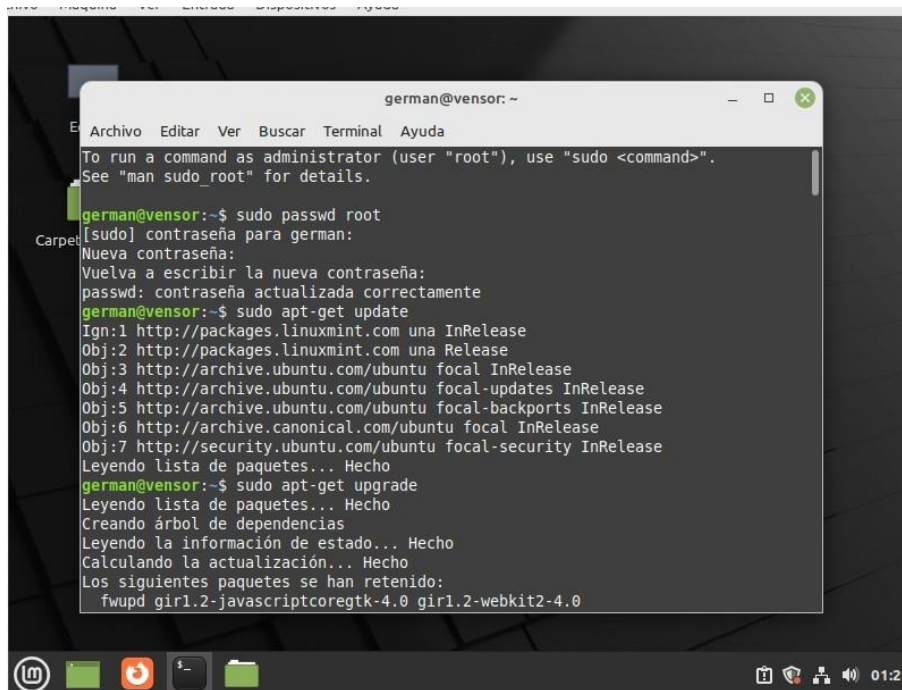
11) En este paso debemos añadir datos como nombre de usuario, equipo y contraseña:



12) Esperamos un momento a que finalice la instalación hasta que nos aparezca este mensaje y reiniciamos:



13) Ya que se haya reiniciado la MV, nos conectamos a una conexión de internet para comprobar el funcionamiento aplicando unos comandos en terminal de actualización de paquetes de software disponibles repositorios oficiales y eliminación de versiones antiguas de paquetes instalados del sistema que ya no se necesitan.



```
german@vensor: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
german@vensor:~$ sudo passwd root  
[sudo] contraseña para german:  
Nueva contraseña:  
Vuelva a escribir la nueva contraseña:  
passwd: contraseña actualizada correctamente  
german@vensor:~$ sudo apt-get update  
Ign:1 http://packages.linuxmint.com una InRelease  
Obj:2 http://packages.linuxmint.com una Release  
Obj:3 http://archive.ubuntu.com/ubuntu focal InRelease  
Obj:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease  
Obj:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease  
Obj:6 http://archive.canonical.com/ubuntu focal InRelease  
Obj:7 http://security.ubuntu.com/ubuntu focal-security InRelease  
Leyendo lista de paquetes... Hecho  
german@vensor:~$ sudo apt-get upgrade  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Calculando la actualización... Hecho  
Los siguientes paquetes se han retenido:  
fwupd gir1.2-javascriptcoregtk-4.0 gir1.2-webkit2-4.0
```

Python

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes.

Python viene preinstalado en la mayoría de distribuciones Linux, y también está disponible como paquete en el resto. Sin embargo, hay determinadas características que puede que quiera usar y no están disponibles en tu paquete de distribución. Puedes compilar fácilmente la última versión de Python de la fuente.

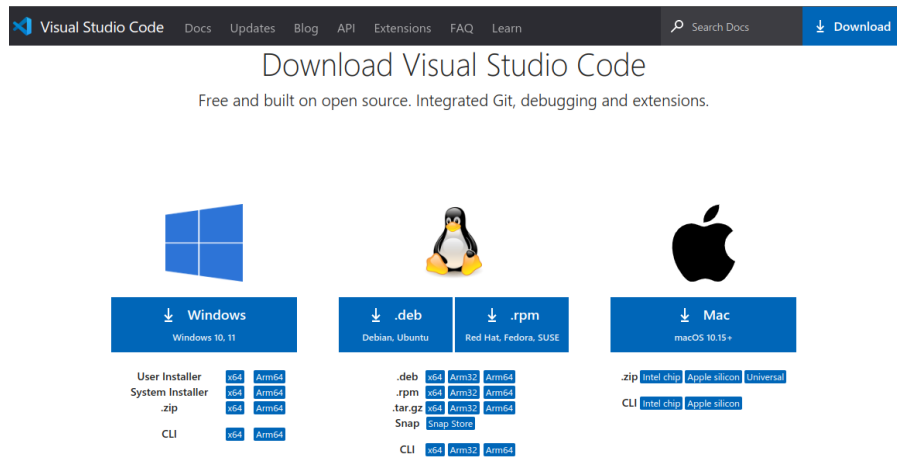
```
german@vensor:~$ python3 --version  
Python 3.10.12
```

Visual Studio Code

Es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación. (Frankier Flores, 2022)

Procedimiento de instalación

1) Primeramente, debemos acceder a la página oficial de VSC y descargar la paquetería de acuerdo al sistema operativo y arquitectura de tu equipo:



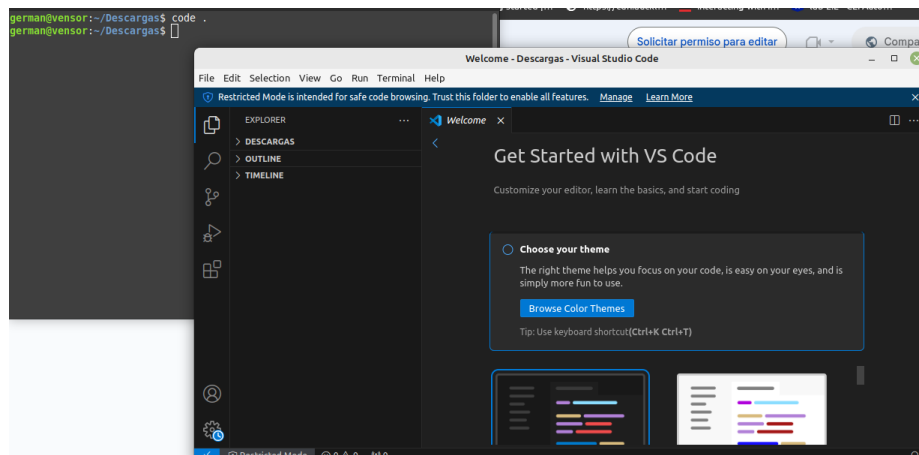
2) Después de descargar la paquetería requerida, debemos actualizar los repositorios con el siguiente comando en terminal Linux:

```
german@vensor:~$ sudo apt update
[sudo] password for german:
Ign:1 https://download.virtualbox.org/virtualbox/debian vanessa InRelease
Ign:2 http://ppa.launchpad.net/yannubuntu/boot-repair/ubuntu jammy InRelease
Ign:3 https://download.docker.com/linux/ubuntu jammy InRelease
```

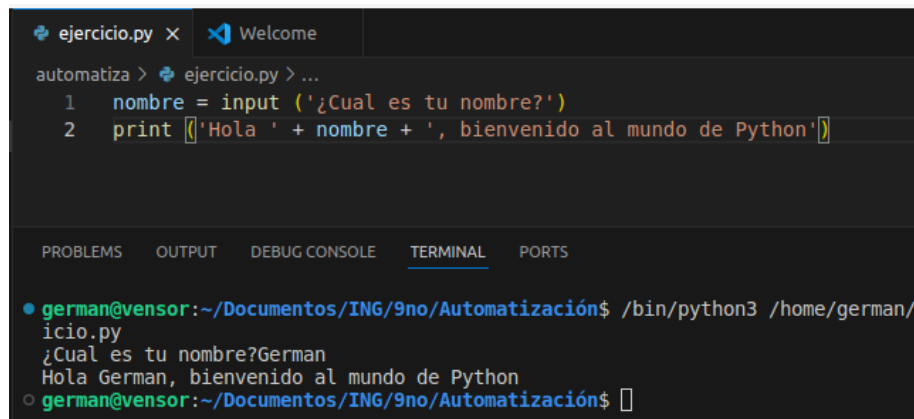
3) Ahora nos dirigimos al directorio donde se encuentra la descarga e instalamos el paquete con el siguiente comando:

```
german@vensor:~$ cd Descargas/
german@vensor:~/Descargas$ sudo dpkg -i code_1.89.1-1715060508_amd64.deb
[sudo] password for german:
(Leyendo la base de datos ... 399452 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar code_1.89.1-1715060508_amd64.deb ...
Desempaquetando code (1.89.1-1715060508) sobre (1.89.1-1715060508) ...
```

4) Ejecutamos el VSC con el comando “code .”, indicando que se abra desde el directorio actual:



- 5) Finalmente, comprobamos su funcionamiento ejecutando un programa en Python:
NOTA: debemos descargar extensiones de lenguaje para poder interactuar con los resultados.



```
ejercicio.py x Welcome
automatiza > ejercicio.py > ...
1 nombre = input ('¿Cual es tu nombre?')
2 print ('Hola ' + nombre + ', bienvenido al mundo de Python')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● german@vensor:~/Documentos/ING/9no/Automatización$ /bin/python3 /home/german/
icio.py
¿Cual es tu nombre?German
Hola German, bienvenido al mundo de Python
○ german@vensor:~/Documentos/ING/9no/Automatización$
```

Git

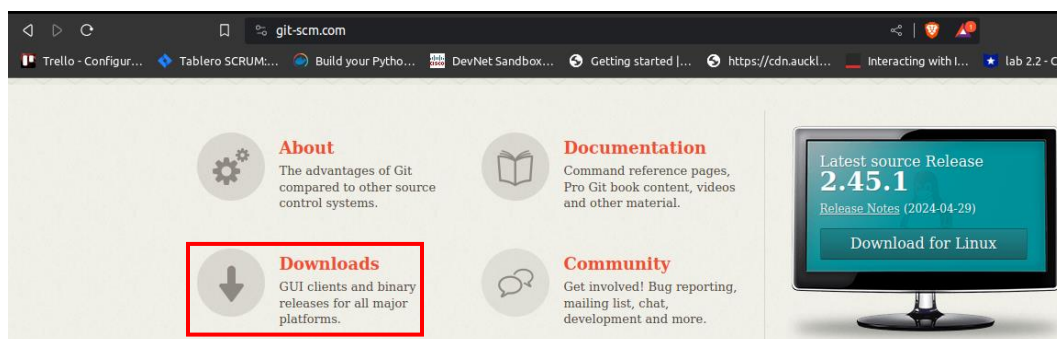
Es el sistema de control de versiones que permite a los desarrolladores administrar cambios en un software a la vez que el proyecto evoluciona.

En el caso de que un desarrollador quisiera trabajar en un proyecto, sería arriesgado realizar cambios sobre el código original. El control de versiones permite duplicar una parte de un proyecto de forma aislada y trabajar sobre ella sin que se modifique el repositorio original.

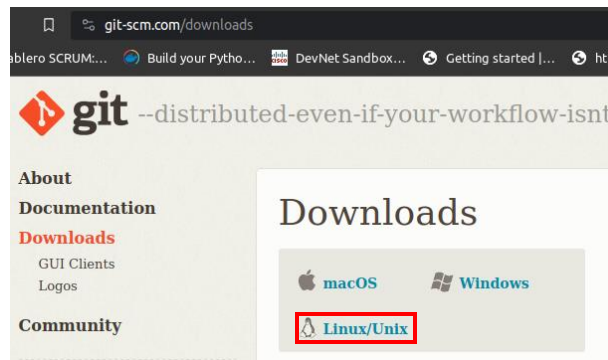
Una vez comprobado que el cambio se ha realizado con éxito, el desarrollador podrá fusionar su ramificación con el proyecto creando una nueva versión del mismo. Esta nueva versión registra los cambios realizados sobre la versión anterior para que se puedan testear por otros desarrolladores. (webempresa)

Procedimiento de instalación

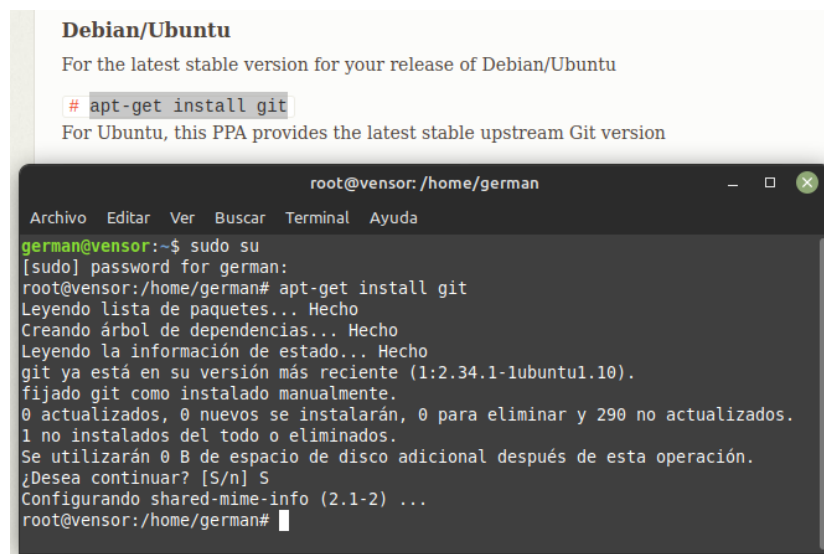
- 1) Primeramente, debemos ingresar a la pagina oficial de Git, para acceder al apartado de descargas:



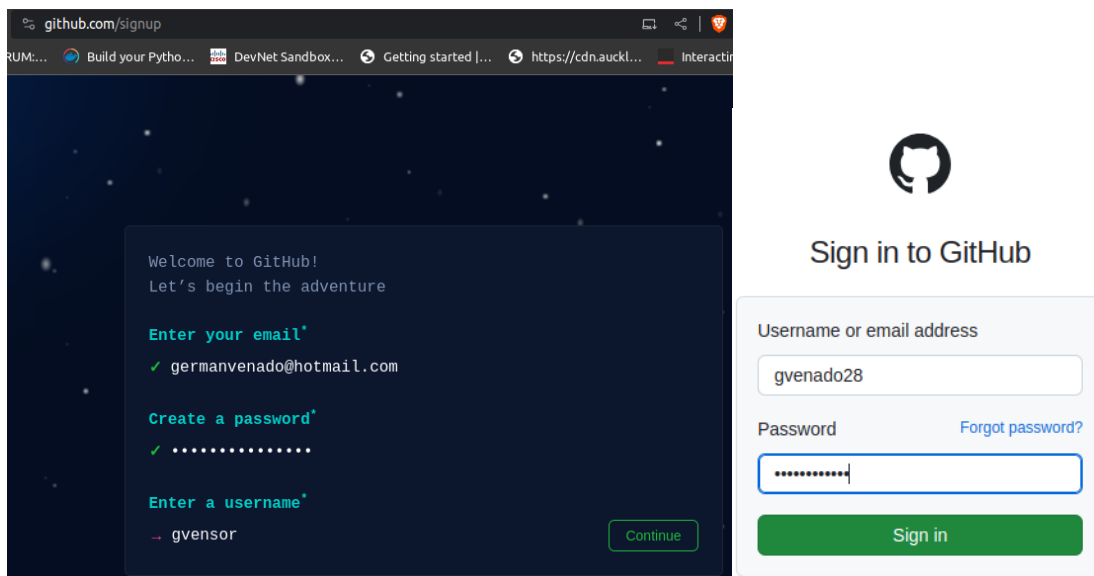
- 2) Seleccionamos el tipo de SO que tenemos para conocer el tipo de descarga:



3) Como Linux Mint es una distribución basada en la comunidad Ubuntu/Debian, ejecuta los mismos comandos para la instalación de Git:



4) Ahora, daremos paso a crear una cuenta y/o registrar cuenta en GitHub. Para crear una cuenta debes disponer de un correo electrónico activo, para poder validar su cuenta de GitHub:



5) Ahora en una terminal iniciaremos el servicio de Git, para empezar a trabajar en el directorio actual y subir lo necesario a GitHub:

```
german@vensor:~/Documentos$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:     git config --global init.defaultBranch <name>
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:     git branch -m <name>
Initialized empty Git repository in /home/german/Documentos/.git/
```

6) Aplicamos los siguientes comandos compartidos en la página de GitHub después de crear un repositorio, el cual creará un archivo llamado “german.txt”, ese archivo será agregado al repositorio de GitHub:

```
german@vensor:~/Documentos$ touch german.txt
german@vensor:~/Documentos$ git add german.txt
german@vensor:~/Documentos$ git commit -m "Interactuando con git"
Author identity unknown

*** Please tell me who you are.

Run

    git config --global user.email "you@example.com"
    git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.
fatal: unable to auto-detect email address (got 'german@vensor.(none)')
```

Docker

Docker es una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. Con Docker, puede implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará.

Procedimiento de instalación

1) Primeramente se deben instalar los paquetes de requisitos previos antes de instalar Docker:

```

german@vensor:~$ sudo apt-get install ca-certificates curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ca-certificates ya está en su versión más reciente (20230311ubuntu0.22.04.1).
curl ya está en su versión más reciente (7.81.0-1ubuntu1.16).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son
necesarios.
cpu-checker ibverbs-providers ipxe-qemu ipxe-qemu-256k-compatible-efi-roms
libcacard0 libdaxctl1 libfdt1 libgfpapi0 libgfrpc0 libgfxdr0 libglusterfs0
libibverbs1 libiscsi7 libndctl6 libpmem1 libpmemobj1 libqrencode4 librados2
librbd1 librdmacm1 libslirp0 libspice-server1 libusbredirparser1
libvirglrenderer1 msr-tools ovmf pass pigz qemu-block-extra qemu-system-common
qemu-system-data qemu-system-gui qemu-system-x86 qemu-utils qrencode seabios
slirp4netns uidmap xclip
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 286 no actualizados.

```

2) Ahora se deben agregar los repositorios de Docker y claves GPG, para facilitar el proceso de instalación y al mismo tiempo utilizar el método de instalación oficialmente compatible:

```

german@vensor:~$ sudo install -m 0755 -d /etc/apt/keyrings
german@vensor:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /e
tc/apt/keyrings/docker.asc
german@vensor:~$ sudo chmod a+r /etc/apt/keyrings/docker.asc

german@vensor:~$ echo \
> "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] h
https://download.docker.com/linux/ubuntu \
> $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
> sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
german@vensor:~$ sudo apt-get update
Ign:1 http://packages.linuxmint.com vanessa InRelease
Obj:2 http://packages.linuxmint.com vanessa Release
Obj:3 http://archive.ubuntu.com/ubuntu jammy InRelease

```

4) Procede a instalar Docker:

```

german@vensor:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-
buildx-plugin docker-compose-plugin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son
necesarios.
cpu-checker ibverbs-providers ipxe-qemu ipxe-qemu-256k-compatible-efi-roms
libcacard0 libdaxctl1 libfdt1 libgfpapi0 libgfrpc0 libgfxdr0 libglusterfs0
libibverbs1 libiscsi7 libndctl6 libpmem1 libpmemobj1 libqrencode4 librados2
librbd1 librdmacm1 libspice-server1 libusbredirparser1 libvirglrenderer1
msr-tools ovmf pass qemu-block-extra qemu-system-common qemu-system-data
qemu-system-gui qemu-system-x86 qemu-utils qrencode seabios uidmap xclip
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
docker-ce-rootless-extras
Paquetes sugeridos:
aufs-tools cgroupfs-mount | cgroup-lite
Se instalarán los siguientes paquetes NUEVOS:
containerd.io docker-buildx-plugin docker-ce docker-ce-cli
docker-ce-rootless-extras docker-compose-plugin
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 286 no actualizados.
Se necesita descargar 51.6 MB/121 MB de archivos.
Se utilizarán 433 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd
64 1.6.31-1 [29.8 MB]

```

5) Finalmente, puedes utilizar la imagen de prueba para verificar que todo funcione como debería:

```
german@vensor:~$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

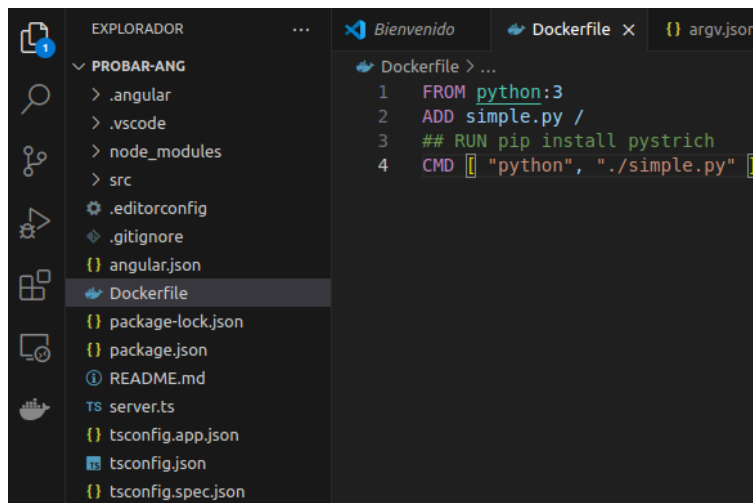
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Ejecución

1) Crear un archivo llamado Dockerfile y pegar la siguiente estructura de código para la creación de un contenedor:



The screenshot shows the Visual Studio Code interface. On the left, the 'EXPLORADOR' (Explorer) sidebar shows a project named 'PROBAR-ANG' with various files and folders. A new file named 'Dockerfile' has been created and is highlighted. The main editor area shows the content of the 'Dockerfile' with the following code:

```
1 FROM python:3
2 ADD simple.py /
3 ## RUN pip install pystrich
4 CMD ["python", "./simple.py"]
```

2) Ejecutar la imagen creada para la ejecución del contenedor:


```

german@vensor:~/Documentos/ING/9no/Automatización/backend$ nano Dockerfile
german@vensor:~/Documentos/ING/9no/Automatización/backend$ nano .dockerignore
german@vensor:~/Documentos/ING/9no/Automatización/backend$ sudo docker build --network=host -t backend .
[+] Building 420.3s (4/9)                                docker:default
=> [internal] load build definition from Dockerfile      0.1s
=> => transferring dockerfile: 209B                      0.0s
=> [internal] load metadata for docker.io/library/node:latest 2.8s
=> [internal] load .dockerignore                        0.0s
=> => transferring context: 92B                          0.0s
=> [1/5] FROM docker.io/library/node:latest@sha256:5e4044ff6001d06e7748e3 417.4s
[+] Building 420.5s (4/9)                                docker:default
=> [internal] load build definition from Dockerfile      0.1s
=> => transferring dockerfile: 209B                      0.0s
=> [internal] load metadata for docker.io/library/node:latest 2.8s
=> [internal] load .dockerignore                        0.0s
=> => transferring context: 92B                          0.0s
=> [1/5] FROM docker.io/library/node:latest@sha256:5e4044ff6001d06e7748e3 417.6s
[+] Building 420.7s (4/9)                                docker:default
=> [1/5] FROM docker.io/library/node:latest@sha256:5e4044ff6001d06e7748e3 417.8s
=> => sha256:3873416e6a335157d669c6193a256dfb289331d669 64.14MB / 64.14MB 261.4s
=> => sha256:64b533fe5853818bcc447638650fa94d56e11e687122ea 7.41kB / 7.41kB 0.0s
=> => sha256:feal432adf0957427b45b0ef8e37cbfe045b7cf8c15 49.58MB / 49.58MB 92.8s
=> => sha256:5e4044ff6001d06e7748e35bfa4f80c73cf5f5a7360a1b 1.21kB / 1.21kB 0.0s
=> => sha256:8d61fb1a55bd92e375915943bd63fceb1174d2c58c08ec 2.00kB / 2.00kB 0.0s
[+] Building 435.9s (4/9)                                docker:default
=> [internal] load build definition from Dockerfile      0.1s
=> => transferring dockerfile: 209B                      0.0s
=> [internal] load metadata for docker.io/library/node:latest 2.8s
=> [internal] load .dockerignore                        0.0s
=> => transferring context: 92B                          0.0s

```

Angular

Es un framework de ingeniería de software de código abierto que se utiliza para crear aplicaciones web de una sola página. Los desarrolladores también lo utilizan para crear menús animados para páginas web HTML.

La arquitectura de una aplicación en Angular se basa en ciertos conceptos fundamentales. Los bloques de construcción básicos son los NgModules, que proporcionan un contexto de compilación para los componentes. Los NgModules recopilan código relacionado en conjuntos funcionales; una aplicación de Angular se define por un conjunto de NgModules. Una aplicación siempre tiene al menos un módulo raíz que permite el arranque y generalmente tiene muchos más módulos de funcionalidad.

Procedimiento de instalación

1) Primeramente, se deben actualizar las paqueterías (apt update), para después instalar el administrador de dependencias (npm) y el módulo node (npm install -g -n) y ejecutar el siguiente paquete:

```

german@vensor:~$ sudo npm i -g @angular/cli
npm WARN deprecated read-package-json@7.0.1: This package is no longer supported. Please use @npmcli/package-json instead.

added 237 packages in 3m

44 packages are looking for funding
  run `npm fund` for details

```


2) Ahora podremos comprobar la versión de NG:

```
german@vensor:~$ ng version
? Would you like to enable autocompletion? This will set up your terminal so
pressing TAB while typing Angular CLI commands will show possible options and
autocomplete arguments. (Enabling autocompletion will modify configuration files in
your home directory.) Yes
Appended `source <(ng completion script)>` to `/home/german/.bashrc`. Restart your t
erminal or run the following to autocomplete `ng` commands:

    source <(ng completion script)
? Would you like to share pseudonymous usage data about this project with the
Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For
more
details and how to change this setting, see https://angular.io/analytics. Yes

Thank you for sharing pseudonymous usage data. Should you change your mind, the fol
lowing
command will disable this feature entirely:
```

```
ng analytics disable --global

Global setting: enabled
Local setting: No local workspace configuration file.
Effective status: enabled

Angular CLI

Angular CLI: 17.3.8
Node: 20.13.1
Package Manager: npm 10.5.2
OS: linux x64

Angular:
...

Package                                Version
-----
@angular-devkit/architect              0.1703.8 (cli-only)
@angular-devkit/core                   17.3.8 (cli-only)
@angular-devkit/schematics             17.3.8 (cli-only)
@schematics/angular                   17.3.8 (cli-only)
```

Ejecución

1) Crear una aplicación con Angular con paquetes automáticos con el siguiente comando:

```

german@vensor:~/Documentos/ING/9no/Automatización$ ng new probar-ang --strict --style=css --routing --skip-tests --no-standalone
? Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? Yes
CREATE probar-ang/README.md (1063 bytes)
CREATE probar-ang/.editorconfig (274 bytes)
CREATE probar-ang/.gitignore (587 bytes)
CREATE probar-ang/angular.json (3499 bytes)
CREATE probar-ang/package.json (1275 bytes)
CREATE probar-ang/tsconfig.json (857 bytes)
CREATE probar-ang/tsconfig.app.json (324 bytes)
CREATE probar-ang/tsconfig.spec.json (273 bytes)
CREATE probar-ang/server.ts (1726 bytes)
CREATE probar-ang/.vscode/extensions.json (130 bytes)
CREATE probar-ang/.vscode/launch.json (470 bytes)
CREATE probar-ang/.vscode/tasks.json (938 bytes)
CREATE probar-ang/src/main.ts (214 bytes)
CREATE probar-ang/src/favicon.ico (15086 bytes)
CREATE probar-ang/src/index.html (295 bytes)
CREATE probar-ang/src/styles.css (80 bytes)
CREATE probar-ang/src/main.server.ts (70 bytes)
CREATE probar-ang/src/app/app-routing.module.ts (245 bytes)
CREATE probar-ang/src/app/app.module.ts (449 bytes)
CREATE probar-ang/src/app/app.component.css (0 bytes)
CREATE probar-ang/src/app/app.component.html (19903 bytes)
CREATE probar-ang/src/app/app.component.ts (211 bytes)
CREATE probar-ang/src/app/app.module.server.ts (318 bytes)
CREATE probar-ang/src/assets/.gitkeep (0 bytes)
✓ Packages installed successfully.
Directory is already under version control. Skipping initialization of git.

```

2) Acceder a la carpeta generada del proyecto para después ejecutar el servicio de Angular:

```

german@vensor:~/Documentos/ING/9no/Automatización$ cd probar-ang/
german@vensor:~/Documentos/ING/9no/Automatización/probar-ang$ ng serve -o
? Would you like to share pseudonymous usage data about this project with the Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
details and how to change this setting, see https://angular.io/analytics. Yes

Thank you for sharing pseudonymous usage data. Should you change your mind, the following
command will disable this feature entirely:

  ng analytics disable

Global setting: enabled
Local setting: enabled
Effective status: enabled
Browser bundles
Initial chunk files | Names | Raw size
polyfills.js | polyfills | 86.27 kB |
main.js | main | 23.06 kB |
styles.css | styles | 95 bytes |
| Initial total | 109.43 kB

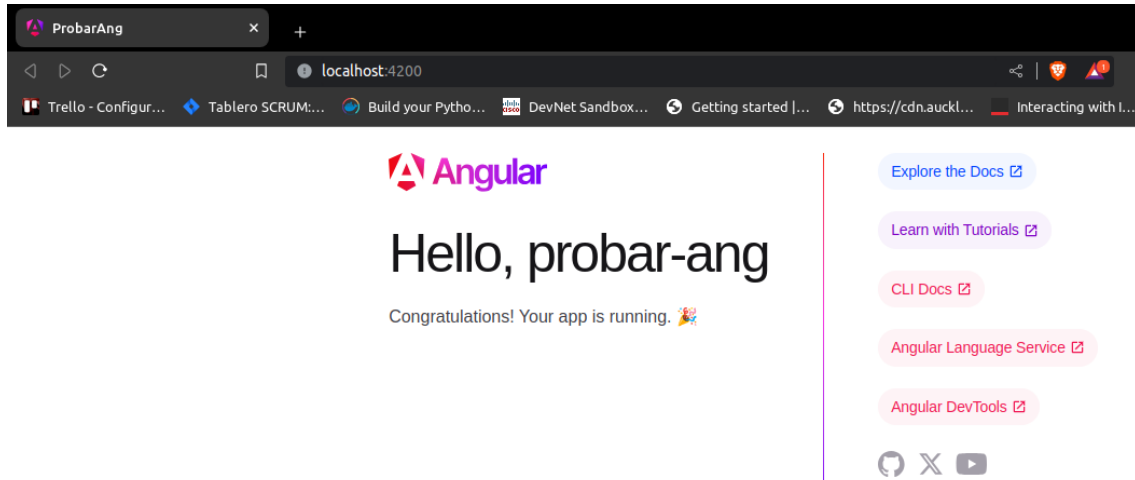
Server bundles
Initial chunk files | Names | Raw size
chunk-T7IUOPKE.mjs | - | 1.70 MB |
polyfills.server.mjs | polyfills.server | 557.68 kB |
main.server.mjs | main.server | 216.52 kB |
chunk-VPSODEBW.mjs | - | 2.51 kB |
render-utils.server.mjs | render-utils.server | 423 bytes |

Lazy chunk files | Names | Raw size
chunk-0TT6LQ5K.mjs | xhr2 | 39.10 kB |

Application bundle generation complete. [42.877 seconds]
Watch mode enabled. Watching for file changes...

```

3) Comprobamos en el navegador con la dirección de localhost y en el puerto 8080 para verificar el estado del servicio:



Alternativas de mejora en la automatización de procesos y servicios tecnológicos

Algunas alternativas de mejora que se podrían aplicar para la automatización en la infraestructura de redes digitales son la siguientes:

- Infraestructura como Código: Herramientas de la Nube permiten definir y provisionar toda la infraestructura de red mediante código, asegurando consistencia y reproducibilidad.
- APIs: Para permitir la integración de diferentes sistemas y servicios, facilitando la comunicación y automatización entre ellos.
- Microservicios: Gestionando servicios individuales sin afectar al sistema completo, mejorando la agilidad y la capacidad de respuesta.
- Contenerización: Empaquetar y desplegar aplicaciones, ofrece varias ventajas clave para la optimización de redes digitales, mejorando eficiencia, flexibilidad y gestión de recursos.

Metodología de desarrollo ágil

Para la ejecución del proyecto solicitado, haré uso y guía de la metodología SCRUM porque se basa en varios beneficios y características que la hacen especialmente adecuada para entornos dinámicos y colaborativos. A continuación, se mencionan algunas razones por las que decidí utilizar la metodología SCRUM:

- Scrum utiliza sprints de corta duración (habitualmente de 1 a 4 semanas) para desarrollar y entregar incrementos de funcionalidad.
- Scrum fomenta la inspección continua del producto en desarrollo y la adaptación de las prácticas y procesos para mejorar la entrega y la satisfacción del cliente.
- Define roles específicos como el Product Owner, Scrum Master y el Equipo de Desarrollo, clarificando responsabilidades y fomentando la colaboración.
- Scrum promueve la entrega continua de incrementos de producto que agregan valor al cliente, permitiendo la validación temprana y la retroalimentación constante.
- Scrum facilita la identificación temprana de riesgos y su gestión proactiva durante el ciclo de vida del proyecto, minimizando impactos negativos en el producto final.
- La metodología Scrum está diseñada para ser ágil y responder rápidamente a cambios en el entorno del proyecto, permitiendo ajustes de prioridades y requisitos basados en nuevas información o requerimientos del negocio.

Historias de usuario

Historia de usuario	Descripción	Criterios de aceptación
Servicio de Angular.	Compartir datos y funcionalidades entre diferentes componentes de la aplicación.	<ul style="list-style-type: none"> • El servicio debe ser inyectable en componentes mediante el sistema de inyección de dependencias de Angular. • El servicio debe ser compatible con las versiones de Angular especificadas en el proyecto. • El servicio debe ser capaz de manejar operaciones de adición y recuperación de usuarios de manera eficiente.
Base de datos para información de usuario de los clientes.	Emplear base de datos para guardar información de los datos personales, credenciales de autenticación, preferencias de usuario, historiales de actividad, entre otros.	<ul style="list-style-type: none"> • El usuario debe poder crear, ver, editar y eliminar cuentas financieras. • Cada cuenta debe tener información detallada, incluyendo nombre, tipo de cuenta, saldo inicial, y moneda.
Autenticación y verificación de usuarios.	Procesos de seguridad para validar la autenticación del usuario y verificación para confirmar la identidad del usuario en base a los datos proporcionados.	<ul style="list-style-type: none"> • Los usuarios deben poder iniciar sesión utilizando su nombre de usuario y contraseña. • Los usuarios deben poder solicitar la recuperación de su contraseña mediante su dirección de correo electrónico. • Los usuarios deben poder activar la autenticación de dos factores (2FA) para añadir una capa adicional de seguridad.

Informes y análisis.	Permitir a los usuarios entender, gestionar y tomar decisiones informadas sobre sus finanzas.	<ul style="list-style-type: none"> • La aplicación debe proporcionar informes financieros detallados, como estados de cuenta, gráficos de gastos e ingresos, y resúmenes de categorías. • Los usuarios deben poder filtrar y exportar estos informes en formatos comunes (por ejemplo, PDF, Excel).
Acceso de administración remota (SSH).	Los administradores de TI pueden realizar su trabajo desde cualquier ubicación con conexión a internet, aumentando la flexibilidad laboral.	<ul style="list-style-type: none"> • Establecer una conexión SSH utilizando las credenciales correctas (nombre de usuario y clave privada). • Todos los datos transmitidos a través de la conexión SSH deben estar cifrados utilizando algoritmos de cifrado seguro (como AES-256).
Balanceo de carga.	Garantizar la disponibilidad, escalabilidad y rendimiento óptimo de aplicaciones financieras.	<ul style="list-style-type: none"> • El balanceo de carga debe ser capaz de escalar horizontalmente para manejar incrementos de tráfico sin degradación significativa del rendimiento.
Medidas de protección contra ataques de fuerza bruta.	Mitigar los intentos repetitivos y automatizados de adivinar credenciales válidas, como contraseñas o claves de acceso.	<ul style="list-style-type: none"> • Los usuarios podrán gestionar y actualizar su información personal, como datos de contacto y preferencias de notificación. • Todas las transacciones financieras deben ser registradas correctamente y

		reflejadas en los estados de cuenta.
Implementación de respaldo y recuperación ante fallas.	Asegurar que, en caso de pérdida de datos debido a errores humanos, fallos de hardware, ataques cibernéticos u otros, se pueda restaurar la información y continuar con las operaciones lo más rápido posible.	<ul style="list-style-type: none"> • Realizar un respaldo completo de la base de datos de la aplicación financiera al menos una vez al día. • Cada respaldo debe ser verificado automáticamente para asegurar que los datos copiados sean idénticos a los datos originales. • Almacenar respaldos en ubicaciones geográficamente separadas para protección contra desastres, asegurando al menos una copia fuera del sitio principal.
Roles y permisos para restringir el acceso a los datos.	Gestionar el acceso y seguridad de los datos de la aplicación financiera.	<ul style="list-style-type: none"> • Todos los roles deben estar claramente definidos en base a las responsabilidades y funciones dentro de la aplicación financiera. • Los usuarios deben tener el mínimo conjunto de privilegios necesarios para realizar sus funciones específicas dentro de la aplicación.
Encriptación de datos sensibles en la base de datos.	Aplicar técnicas criptográficas para proteger la información confidencial almacenada, de modo que solo las personas autorizadas puedan acceder a ella.	<ul style="list-style-type: none"> • Utilizar algoritmos criptográficos estándar y recomendados, como AES-256 para el cifrado simétrico y RSA o ECC para el cifrado asimétrico.

		<ul style="list-style-type: none"> • Las claves de encriptación deben almacenarse de manera segura, utilizando métodos como el almacenamiento en hardware seguro (HSM) o el uso de servicios de gestión de claves (KMS) ofrecidos por proveedores de la nube. • Las claves de encriptación deben estar protegidas contra accesos no autorizados y deben ser accesibles solo para los sistemas y usuarios autorizados.
--	--	---

Backlog

El siguiente hipervínculo se encuentra un enlace dónde podrá visualizar el Backlog desarrollado en la plataforma de Jira, teniendo organización profesional de tareas del proyecto:

[Ver Backlog](#)

Sprint Planning

- Objetivo del Sprint: Conocer e interactuar con servicios en la nube para agilizar los procesos con menores recursos informáticos.
- Descomposición y estimación del trabajo:
Definición de
- Creación del Sprint Backlog:

▼ Tablero Sprint 1 Añadir fechas (Visibles: 3 de 8 incidencias)		29	0	0	Iniciar sprint
BDF-12	Definir arquitectura de Software para la aplicación	CAPA DE LÓGI...	TAREAS POR H...	10	GS
BDF-11	Conocer e implementar microservicios en nube	CAPA DE LÓGI...	TAREAS POR H...	10	GS
BDF-5	Implementar el balanceo de carga	CAPA DE LÓGI...	TAREAS POR H...	9	GS

Comunidades de colaboración, desarrollo y automatización de redes

Las comunidades de colaboración, desarrollo y automatización de redes juegan un papel crucial en la evolución y el avance de las tecnologías de red modernas. Estas comunidades reúnen a profesionales, entusiastas, desarrolladores y expertos en redes para compartir conocimientos, colaborar en proyectos, desarrollar herramientas y automatizar procesos relacionados con redes.

Importancia y Beneficios

- Aprendizaje Continuo: Facilitan el aprendizaje continuo y la actualización sobre las últimas tecnologías y mejores prácticas en redes.
- Innovación Colaborativa: Fomentan la colaboración entre profesionales y organizaciones para desarrollar soluciones innovadoras y resolver desafíos complejos en redes.
- Desarrollo Profesional: Proporcionan oportunidades para desarrollar habilidades técnicas, mejorar la eficiencia operativa a través de la automatización, y prepararse para certificaciones y roles avanzados en redes.

En resumen, estas comunidades son fundamentales para el desarrollo profesional y la evolución de las tecnologías de red, ofreciendo un espacio vital para la colaboración, el aprendizaje y la innovación en la automatización y gestión de redes.

Mapa mental



CONCLUSIÓN

La automatización de infraestructura de redes representa un avance crucial en la gestión moderna de sistemas, ofreciendo beneficios significativos en términos de eficiencia operativa, agilidad y reducción de errores. Al adoptar herramientas y prácticas automatizadas, las organizaciones pueden optimizar el tiempo de implementación de cambios, mejorar la consistencia de configuraciones y fortalecer la seguridad de sus redes. Esto no solo permite a los equipos de IT centrarse en tareas estratégicas y de alto valor, sino que también facilita la adaptación rápida a las demandas cambiantes del negocio y del entorno tecnológico. En resumen, la automatización de infraestructura de redes no solo es una tendencia, sino una necesidad creciente para mantenerse competitivo y ágil en un mundo digitalmente conectado y dinámico.

BIBLIOGRAFÍA

Flores, F., (2022), OpenWebinars, Qué es Visual Studio Code y qué ventajas ofrece, Recuperado de: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>

García, F., (2022), arsys, Linux Mint, una distribución diseñada para llevar la experiencia Linux al escritorio, Recuperado de: <https://www.arsys.es/blog/linux-mint-una-distribucion-disenada-para-llevar-la-experiencia-linux-al-escritorio>

s/a, (2019), RedHat, ¿Qué es la automatización de la infraestructura?, Recuperado de: <https://www.redhat.com/es/topics/automation/what-is-infrastructure-automation>

s/n, s/f, aws, ¿Qué es Docker?, Recuperado de: <https://aws.amazon.com/es/docker/>

s/n, s/f, webempresa, ¿Qué es GitHub y para qué sirve?, Recuperado de: <https://www.webempresa.com/hosting/que-es-github.html>

Scarpatti J., (2018), ComputerWeekly.es, Automatización de redes, Recuperado de: <https://www.computerweekly.com/es/definicion/Automatizacion-de-redes>

Valio, (2022), LinkedIn, Por qué es tan popular Linux en el mundo de los desarrolladores, Recuperado de: <https://es.linkedin.com/pulse/por-qu%C3%A9-es-tan-popular-linux-en-el-mundo-de-los-desarrolladores->